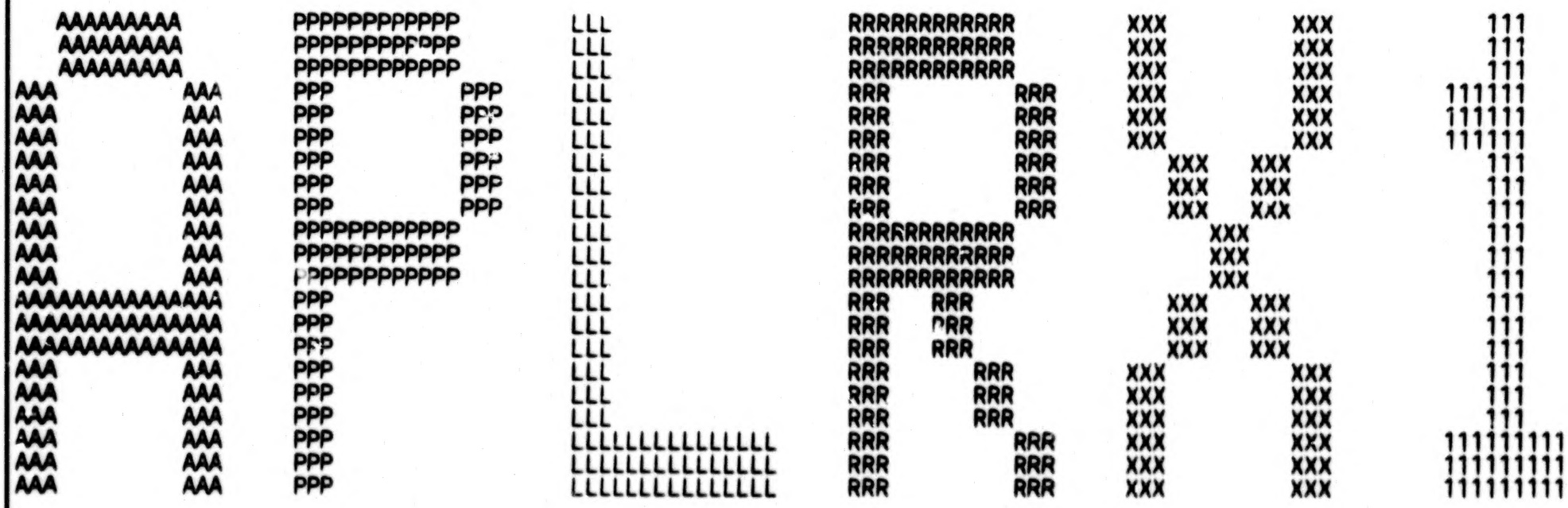
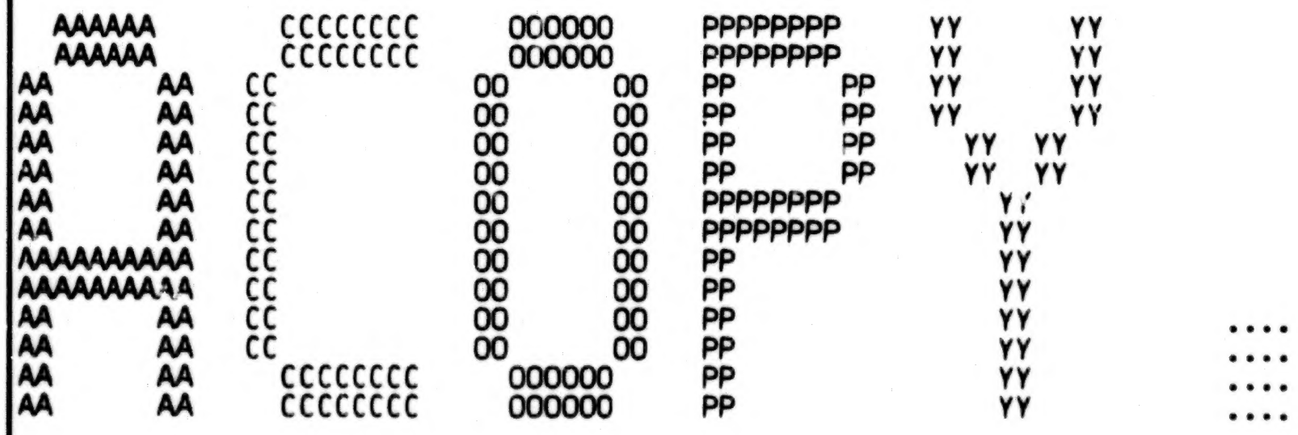
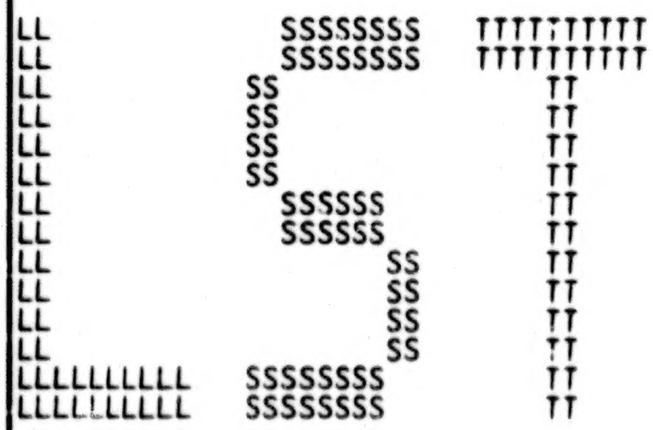
TITLE1**AH-J618A-SC
TITLE2**APL-11/RSX V2.0 SRC MCRF



FILEID**ACOPY





.MAIN. MACRO M1113 08-FEB-80 08:56
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
     2-    29    TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
     2-    33    AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    ACOPY           COPY ARRAY VALUES
```

```
ACOPY   MACRO M1113  08-FEB-80 08:56  PAGE 1


        1           000000                  RT11    =  0            ;NOT RT-11
        2           000000                  RSTS    =  0            ;NOT RSTS/E
        3           000115                  RSX     = 'M            ;RSX-11/M
        4
        5           000000                  EIS     =  0            ;NO EIS
        6           000000                  FIS     =  0            ;NO FIS
        7           000000                  FPP     =  0            ;NO FPP
        8
        9           000004                  FPPLEN  =  4.           ;SINGLE PRECISION
       10
       11           000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  ACOPY
                                .IDENT  \V02.03\

                                .SBTTL  ACOPY           COPY ARRAY VALUES

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       ACOPY is used by various routines to copy all
                                ;       or part of an array (be it numeric or literal).
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    ACOPY
                                ;
                                ; INPUTS:
                                ;       R0      - Number of elements (not words!) to copy
                                ;       R1      - Destination pointer (not pointer to block header)
                                ;       R2      - Pointer to source block header
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       The data in the source array is copied to the
                                ;       destination array.
                                ;--

000000                          .PSECT  CODE

000000  005700                 ACOPY:: TST     R0              ;ANYTHING TO COPY?
000002  001430                         BEQ     5$              ;NO!
000004                          PUSH    R2              ;SAVE PTR TO SOURCE
000006  116202  000003          MOVB    A.BITS(R2),R2   ;GET BITS
000012  042702  177771          BIC     #^C<A.NUM>,R2   ;CLEAR GARBAGE
000016  005762  000044'         TST     SIZTAB(R2)      ;SOURCE ARRAY IN WORDS?
000022  001005                  BNE     2$              ;YES
000024  013602                  MOV     @(SP)+,R2       ;ELSE COPY BYTE BY BYTE
000026  112221                  1$:     MOVB    (R2)+,(R1)+     ;COPY ONE BYTE
000030                          SOB     R0,1$           ;COPY ENTIRE ARRAY
000034  000413                  BR      5$              ;ALL DONE
000036                          2$:     ASH     SIZTAB(R2),R0   ;MULTIPLY BY PRECISION
000052  006200                  ASR     R0              ;CONVERT TO WORD COUNT
000054  013602                  MOV     @(SP)+,R2       ;POINT TO FIRST SOURCE ELEMENT
000056  012221                  4$:     MOV     (R2)+,(R1)+     ;COPY ARRAY
000060                          SOB     R0,4$           ;VERY FAST!
000064                          5$:     RETURN
```

```
ACOPY   MACRO M1113  08-FEB-80 08:56  PAGE 46
ACOPY           COPY ARRAY VALUES

      1         000001                          .END
```

ACOPY MACRO M1113 08-FEB-80 08:56 PAGE 46-1
SYMBOL TABLE

```
ACO     000000RG    020 CHRFRO= 000135      C.BLOK= 000064      D.DECD= 000002      EIS    = 000000
AC0    =%000000         CHRLBR= 000073      C.BLSZ= 000140      D.DIV = 000000      ELESIZ  005132R     025
AC1    =%000001         CHRLPR= 000072      C.BUFF= 177760      D.DMD = 000010      ENDCOR  000004R     025
AC2    =%000002         CHRMNS= 000137      C.BUFL= 177762      D.DQAD= 177777      EOF    = 104475
AC3    =%000003         CHRMUL= 000075      C.CCNT= 177766      D.DROP= 000002      ERRMAX= 000166
AC4    =%000004         CHRNEG= 000100      C.CPTR= 177764      D.ENCD= 000002      ERRNUM  004772R     025
AC5    =%000005         CHRPDL= 000271      C.CZ  = 001000      D.EQ  = 000000      ERRPTR  004770R     025
ALLINT  003274R     025 CHRPLU= 000055      C.DIG = 000040      D.EXPD= 000002      ERR0    004774R     025
ARYRET  003276R     025 CHRPWR= 000120      C.FASW= 020000      D.FGET= 000005      EXEBUF  005054R     025
ASCQUO= 000042          CHRQST= 000121      C.FLEN= 000010      D.FMT = 000004      E$CT    005056R     025
A.ALEN= 000004          CHRQUO= 000113      C.FLSH= 004000      D.FPNT= 000005      E$PT    005060R     025
A.ARG = 000040          CHRRBR= 000047      C.FRC = 000001      D.FPUT= 000005      FA.APD= 000100
A.ASB = 000200          CHRRPR= 000042      C.IDEN= 000010      D.GE  = 000000      FA.CRE= 000010
A.BITS= 000003          CHRSEM= 000074      C.NAME= 000102      D.GT  = 000000      FA.DLK= 001000
A.CON = 000020          CHRSMO= 000112      C.NOBF= 000400      D.IBM = 000002      FA.ENB= 100000
A.DIMS= 000002          CHRSPA= 000040      C.NUM = 000020      D.IND = 000002      FA.EXC= 002000
A.DIM1= 000006          CHRUPA= 000131      C.OP  = 000004      D.INNR= 000013      FA.EXT= 000004
A.NUM = 000006          CHR.BS= 000010      C.OPN = 010000      D.INTR= 000006      FA.NSP= 000100
A.PTR = 000000          CHR.CC= 000003      C.RDLN= 177776      D.LE  = 000000      FA.NSR= 000200
A.TMP = 000010          CHR.CR= 000015      C.REWT= 002000      D.LOG = 000001      FA.POS= 010000
BADDIR= 104463          CHR.CZ= 000032      C.RND = 000010      D.LT  = 000000      FA.RD = 000001
BADERR= 104437          CHR.GT= 000046      C.SBIT= 177756      D.MAX = 000000      FA.RWD= 004000
BLZBUB  005116R     025 CHR.LF= 000012      C.SPFX= 000100      D.MEM = 000002      FA.SEQ= 040000
BRSELF  005046R     025 CHR.LT= 000043      C.SPGL= 000200      D.MIN = 000000      FA.SHR= 000040
CCNT    005072R     025 CHR.SI= 000017      C.SPLX= 000001      D.MUL = 000000      FA.TMP= 000020
CHANEX  003060R     025 CHR.SL= 000057      C.SPRS= 000002      D.NE  = 000000      FA.WCK= 020000
CHAN00  000256R     025 CHR.SO= 000016      C.SRC = 177774      D.NN  = 000000      FA.WRT= 000002
CHAN01  000330R     025 CHR.TB= 000011      C.SRCP= 000024      D.NR  = 000000      FD.BLK= 000010
CHAN02  000512R     025 CHR.UL= 000106      C.TT  = 000004      D.OR  = 000000      FD.CCL= 000002
CHAN03  000674R     025 CHR..A= 000141      C.WIDC= 177770      D.PLUS= 000000      FD.COM= 020000
CHAN04  001056R     025 CHR..E= 000145      C.WIDR= 177772      D.PSUB= 000006      FD.CR = 000002
CHAN05  001240R     025 CHR..S= 000163      DEBUG = 000000      D.PWR = 000001      FD.DIR= 000010
CHAN06  001422R     025 CHR..T= 000164      DEFERR= 104405      D.QQAD= 000000      FD.FTN= 000001
CHAN07  001604R     025 CHR..U= 000165      DEVIL   005114R 025 D.QUAD= 000000      FD.F11= 040000
CHAN08  001766R     025 CHR..Z= 000172      DEVNTF= 104470      D.RES = 000000      FD.INS= 000010
CHAN09  002150R     025 CHR..0= 000060      DIGITS  000006R 025 D.RHO = 000000      FD.ISP= 002000
CHAN10  002332R     025 CHR..1= 000061      DIMAX = 000020      D.ROT = 000003      FD.MNT= 100000
CHAN11  002514R     025 CH.AND= 000001      DIMBLK  003112R 025 D.SCON= 000006      FD.OSP= 004000
CHAN12  002676R     025 CMDBUF  003250R 025 DIMPTR  003114R 025 D.SDIF= 000006      FD.PLC= 000004
CHAN13  000022R     023 CMDMAX= 000010      DIMTYP  003116R 025 D.SUB = 000000      FD.PRN= 000004
CHAN14  000204R     023 CNTSIZ  005130R 025 DIRFUL= 104466      D.SUBS= 000006      FD.PSE= 010000
CHNOPN= 104471          COMBIN  005076R 025 DOMERR= 104417      D.TAKE= 000002      FD.RAH= 000001
CHRAND= 000051          COMCNT  005126R 025 DPTERR= 104416      D.TRAN= 000004      FD.RAN= 000002
CHRBEL= 000007          COMERR= 104425      DYDCNT  003224R 025 D.UNON= 000006      FD.REC= 000001
CHRBKR= 000133          COMPTR  005122R 025 D.AND = 000000      D.$MAX= 000000      FD.RWM= 000001
CHRBOL= 000200          COMRTN  003156R 025 D.BXCT= 000007      D.$MIN= 000000      FD.SDI= 000020
CHRBOX= 000114          COMSKP  005124R 025 D.BXIO= 000007      D.$MUL= 000000      FD.SQD= 000040
CHRCMA= 000054          CORPTR  005120R 025 D.BXNL= 000007      D.$PLU= 000000      FD.TTY= 000004
CHRCOL= 000076          CS.ASS= 000200      D.BXPP= 000007      D.$RE$= 000000      FD.WBH= 000002
CHRDEL= 000107          CS.BOO= 000010      D.BXPW= 000007      D.$SUB= 000000      FF.CHR= 000005
CHRDIV= 000053          CS.FLO= 000020      D.BXRL= 000007      D.1COM= 000001      FF.NV = 000003
CHRDLR= 000176          CS.IDN= 000040      D.CIRC= 000001      D.1EXP= 000002      FF.POE= 000002
CHRDLT= 000110          CS.ONE= 000100      D.COMB= 000001      D.1ROT= 000003      FF.RWD= 000001
CHRDOT= 000056          CTEPS   000152R 025 D.COMP= 000001      EDICHR  005063R 025 FF.RWF= 000006
CHREP = 000105          CTMAX   000212R 025 D.CONT= 000006      EDICNT  005064R 025 FF.SPC= 000004
CHRERR= 104431          CTRMAX  000172R 025 D.CTLM= 000000      EDIFLG  005062R 025 FHEADR  005012R     025
CHRFIL= 000177          C.ALOC= 000162      D.DEAL= 000003      EDIT   = 000003     FHRET   005040R     025
```

ACOPY MACRO M1113 08-FEB-80 08:56 PAGE 46-2
SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FIS   = 000000          F.FNDF= 000010          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FIXPTR  003074R 025     F.FUN = 000010          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FLAGS   004764R 025     F.FVER= 000120          INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FLCBUF  003250R 025     F.HEAD= 000006          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLINE   005014R 025     F.HIBK= 000004          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLOEXP  003270R 025     F.LBL = 000020          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLONEG  003272R 025     F.LINE= 000012          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLSWTB  000222R 025     F.LOC = 000004          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FNINSI  005044R 025     F.LUN = 000042          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.RD = 00000?          F.MON = 000002          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FPMODE  000042R 025     F.NREC= 000030          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPP   = 000000          F.OPN = 000010          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FREBLK  007400R 025     F.PREV= 000000          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FUZZ    000142R 025     F.PRNT= 100000          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FXPTR1  003076R 025     F.PROT= 010000          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVI = 000200          F.RATT= 000001          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F.ACTL= 000076          F.READ= 004000          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BDB = 000070          F.RET = 177772          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BGPC= 000057          F.RSIZ= 000002          LENERR= 104412      M.GDUP= 000003      ND.SEX= 000000
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R 025 M.GDWN= 000003      ND.SES= 000020
F.BKDN= 000026          F.R2  = 000014          LEXPTR  003100R 025 M.GETI= 000006      ND.SNM= 000000
F.BKDS= 000020          F.R3  = 000016          LFTOP   005066R 025 M.IBM = 000003      ND.SNS= 000040
F.BKEF= 000050          F.R4  = 000020          LINC    005102R 025 M.IDN = 000000      ND.SPN= 001400
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKST= 000024          F.SPDV= 000072          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BR  = 000020          F.STBK= 000036          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.CHR = 000075          F.STK = 000022          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CODE= 000010          F.STP = 000001          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DSPT= 000044          F.TRC = 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.EFBK= 000010          F.URBD= 000020          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFN = 000050          F.VAL = 177776          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.END = 002000          F.VBN = 000064          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.ERR = 000052          GNOME   003102R 025     L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
F.FACC= 000043          HASHC   005136R 025     MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FFBY= 000014          HASHTB  005140R 025     MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
F.FLAG= 000002          HSHMSK= 177601          MONBLK  003120R 025 M.RDCT= 000011      NUMQFL  003234R 025
F.FLB = 000100          HSHSIZ= 000200          MONCVT  003132R 025 M.REVS= 000003      NUMRIG  003231R 025
F.FNAM= 000110          IASH  = ****** GX       MONLEN  003130R 025 M.ROLL= 000000      NUMWID  003226R 025
```

SYMBOL TABLE

```
NUMZFL  003235R      025 RARGBL  003204R      025 R.OFF  = 000000          SY.CLR= 000040          T.LIT  = 000020
NYIERR= 104432           RARGCV  003220R      025 R.PREV= 000004          SY.ERP= 000020          T.MON  = 000010
N.BLSZ= 000036           RARGFL  003202R      025 R.PTR  = 000000          SY.NOU= 000100          T.NIL  = 000004
N.DESC= 000020           RARGLE  003216R      025 R.SEQ  = 000003          SY.OPN= 000002          T.NUM  = 002000
N.DEV = 000032           RARGLN  003210R      025 R.VAR  = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034           RARGPT  003206R      025 R6     =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DID = 000024           RARGSF  003214R      025 R7     =%000007          SY.SVC= 000004          T.SOP  = 000002
N.DVNM= 000032           RARGTY  003212R      025 SI     =%000005          SY.SVL= 000200          T.SPOP= 000040
N.EXT = 000014           RARGVA  003222R      025 SIDAMN= 104433           S.AB   = 177776          T.TMP  = 000200
N.FID = 000000           RBASE   005074R      025 SIFRAM  003314R      025 S.ASB  = 100000          T.VAL  = 100000
N.FNAM= 000006           RCOUNR  005112R      025 SISTAK  006350R      025 S.BFHD= 000020          T.VAR  = 001000
N.FTYP= 000014           RCOUNT  005110R      025 SITOP   006346R          S.BITS= 000002          UPPER  = 177400
N.FVER= 000016           REDCNT  005070R      025 SITP    007144R      025 S.CHR1= 000003          UTNAME  000344R      023
N.NAME= 000006           REDIGN  005066R      025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.NEXT= 000022           REDOFF  005072R      025 SIZTAB  000044R      025 S.FDB  = 000140          VARBLE  007400R      025
N.PPN  = 000024          REENTR  003060R      025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R      025
N.STAT= 000020           REGTYP= 000000           SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R      025
N.UNIT= 000034           RESBIT  003142R      025 SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
OBUFLN= 001000           RESCVT  003152R      025 SI.VAL= 177776           S.FNBW= 000017          WIDTH   000072R      025
OCHAN   005052R      025 RESERR= 104440           SKPSIZ  005134R      025 S.FNTY= 000004          WROUND  000020R      025
OMODE   000402R      023 RESLEN  003150R      025 SPHOLD  000004R      024 S.FTYP= 000002          WSDATE  000136R      025
OPTMP0  005066R      025 RESPTR  003140R      025 SPINIT  000002R      024 S.LEN  = 000002          WSFULL= 104403
OPTMP1  005070R      025 RESSFT  003146R      025 STKINC= 000774           S.LINK= 177774          WSNAME  000074R      025
OPTMP2  005072R      025 RESTYP  003144R      025 STOP   = 104420           S.NFEN= 000020          WSTIME  000132R      025
OPTMP3  005074R      025 RESULT  003136R      025 SUBCNT  003104R      025 S.TYPE= 140000          $FIS   = ****** G
OPTMP4  005076R      025 RESVAL  003154R      025 SUBDCT  003106R      025 S.VAL  = 000000          $FPP   = ****** G
OPTMP5  005100R      025 RINC    005104R      025 SUBPTR  003110R      025 TKDIMS  005072R      025 $FPPLN= ****** G
OPTMP6  005102R      025 RLTBAS  005066R      025 SYMB1   005344R      025 TKFRAM  005066R      025 $MACHN= ****** G
OPTMP7  005104R      025 ROTINC  005070R      025 SYMF    005340R      025 TKLEFT  005070R      025 $RSTS  = ****** G
OPTMP8  005106R      025 ROUND   000010R      025 SYMPT1  005346R      025 TMODE   000403R      023 $RSX   = ****** G
OPTMP9  005110R      025 ROVER   003312R      025 SYMTAB  005352R      025 TMPTAG= 000166          $RT11  = ****** G
OPTM10  005112R      025 RSTS   = 000000           SYMTBF= 104404           TTYBUF  004140R      025 .BOOLE= 000002
OPTM14  005026R      025 RSX    = 000115           SYMTOP  005350R      025 TTYLEN= 000622           .CHARA= 000000
ORIGIN  000030R      025 RTOP    005070R      025 SYMTP   006272R      025 TURKEY= 104436           .DEF.  = 000002
OUTPTR  000066R      025 RT11   = 000000           SYMURN= 000054           TYPE   = 000166          .FLOAT= 000006
OVROPR= 000000           R.BITS= 000003           SYNERR= 104407           T.BOL  = 020000          .FSRCB= ****** G
PATCH   007344R      025 R.FIX  = 000001           SYSERR= 104402           T.CON  = 004000          .INTEG= 000004
PRECSN  000052R      025 R.LCT  = 000002           SYSTAT  000000R      024 T.DIM  = 000100          .MOLUN  000404R      023
RANDAC  003064R      025 R.LN   = 000002           SY.BAD= 000001           T.FUN  = 010000          ...GBL= 000000
RANDY   003062R      025 R.LNUM= 000010           SY.BRK= 100000           T.LIST= 000400           ...TPC= 000020
RANERR= 104411           R.NEXT= 000006           SY.BS  = 000010

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTR   000000  005
MONTB   000000  006
DYDSTB  000000  007
LOWTR   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
ACOPY   MACRO M1113  08-FEB-80 08:56  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000066      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  1298 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:42
ACOPY,DM1:[305,301]ACOPY/-SP=APLRX1,APLDEF,ACOPY,END
```

FILEID**APLDX1

```
  AAAAAA      PPPPPPPP      LL            DDDDDDDD      XX      XX       11
  AAAAAA      PPPPPPPP      LL            DDDDDDDD      XX      XX       11
AA      AA    PP      PP    LL            DD      DD    XX      XX     1111
AA      AA    PP      PP    LL            DD      DD    XX      XX     1111
AA      AA    PP      PP    LL            DD      DD      XX  XX         11
AA      AA    PP      PP    LL            DD      DD      XX  XX         11
AA      AA    PPPPPPPP      LL            DD      DD        XX           11
AA      AA    PPPPPPPP      LL            DD      DD        XX           11
AAAAAAAAAA    PP            LL            DD      DD      XX  XX         11
AAAAAAAAAA    PP            LL            DD      DD      XX  XX         11
AA      AA    PP            LL            DD      DD    XX      XX       11          ....
AA      AA    PP            LL            DD      DD    XX      XX       11          ....
AA      AA    PP            LLLLLLLLLL    DDDDDDDD      XX      XX     111111        ....
AA      AA    PP            LLLLLLLLLL    DDDDDDDD      XX      XX     111111        ....


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        T
LL                    SS        T
LLLLLLLLLL    SSSSSSSS
LLLLLLLLLL    SSSSSSSS
```

APLDF MACRO M1113 08-FEB-80 08:58
TABLE OF CONTENTS

	1		APLDEF
	8		A P L
	9		
	10		A PROGRAMMING LANGUAGE
	11		
	12		STEVE POULSEN & DAVE VANN
	13		11-Jul-76
	14		ROBERT GOTTLIEB & PETER HAYNES
	15		2-Feb-79
	16		
	17		V2
	18		
	19		
	20		COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
	21		DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
	22		
	23		
	24		THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
	25		COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
	26		COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
	27		BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
	28		USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
	29		TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
	30		DEC.
	31		
	32		THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
	33		AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
	34		CORPORATION.
	35		
	36		DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
	37		SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
	38		
	39		
4-	46	DEFINITIONS	ASSEMBLY PARAMETERS
5-	139	DEFINITIONS	OPERATOR OVERLAYS
6-	268	DEFINITIONS	REGISTERS
7-	331	DEFINITIONS	MEMORY ORGANIZATION
8-	377	DEFINITIONS	ORDERING PSECTS
9-	426	DEFINITIONS	SI PARSE FRAMES
10-	474	DEFINITIONS	TOKEN DESCRIPTION BITS
11-	515	DEFINITIONS	SI EXECUTION FRAMES
12-	580	DEFINITIONS	SPECIAL PARSE CHARACTERS
13-	607	DEFINITIONS	CHRTAB BITS
13-	624	DEFINITIONS	SYSTEM STATUS BITS
14-	640	DEFINITIONS	F-FRAMES
15-	724	DEFINITIONS	FUNCTION LINE BLOCKS
17-	807	DEFINITIONS	ARRAY STORAGE BLOCK
18-	869	DEFINITIONS	LIST STORAGE BLOCK
19-	911	DEFINITIONS	INDEXING STACK FRAME
20-	940	DEFINITIONS	SYMBOL TABLE ENTRY
21-	969	DEFINITIONS	CHANNEL BLOCK
23-	1085	DEFINITIONS	RSX DATA
24-	1110	DEFINITIONS	NAME BLOCKS
34-	1612	DEFINITIONS	MACROS
36-	1905	DEFINITIONS	APL CHARACTERS
37-	1967	DEFINITIONS	ERRORS

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 1


                                                
     1          000000          RT11    =  0         ;NOT RT-11
     2          000000          RSTS    =  0         ;NOT RSTS/E
     3          000115          RSX     = 'M         ;RSX-11/M
     4
     5          000000          EIS     =  0         ;NO EIS
     6          000000          FIS     =  0         ;NO FIS
     7          000000          FPP     =  0         ;NO FPP
     8
     9          000004          FPPLEN  =  4.        ;SINGLE PRECISION
    10
    11          000000          DEBUG   =  0         ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .TITLE  APLDF
                                .IDENT  \V02.03\

                                ;THIS FILE IS USED TO DEFINE THE APL SYSTEM INTERNAL VARIABLES

000000                          DEFINE=0                        ;DEFINE THE SYMBOL 'DEFINE'
000000                          LISTDF=0                        ;DEFINE THE SYMBOL 'LISTDF'
```

```
                        .SBTTL  APLDEF
                        .IDENT  \V02.03\

                        .IIF NDF LISTDF, .NLIST SEQ,LOC,BIN,SRC
                        .ENABL  LC, GBL
                        .NLIST TTM,CND
                        .SBTTL                          A P L
                        .SBTTL
                        .SBTTL                  A PROGRAMMING LANGUAGE
                        .SBTTL
                        .SBTTL                STEVE POULSEN & DAVE VANN
                        .SBTTL                        11-Jul-76
                        .SBTTL              ROBERT GOTTLIEB & PETER HAYNES
                        .SBTTL                        2-Feb-79
                        .SBTTL
                        .SBTTL                          V2
                        .SBTTL
                        .SBTTL
                        .SBTTL  COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
                        .SBTTL  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                        .SBTTL
                        .SBTTL
                        .SBTTL  THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
                        .SBTTL  COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
                        .SBTTL  COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
                        .SBTTL  BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
                        .SBTTL  USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
                        .SBTTL  TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
                        .SBTTL  DEC.
                        .SBTTL
                        .SBTTL  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                        .SBTTL  AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                        .SBTTL  CORPORATION.
                        .SBTTL
                        .SBTTL  DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
                        .SBTTL  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
                        .SBTTL
                        .SBTTL


000003                  EDIT=   03
000002                  VRSION= 02
000166                  TYPE=   'v      ;'x' EXPERIMENTAL       'v' FINAL FOR VERSION END EDIT
```

```
                                .SBTTL  DEFINITIONS     ASSEMBLY PARAMETERS


                                ;    IF APL IS BEING DEBUGGED, 'DEBUG' SHOULD BE SET TO 1, WHICH
                                ;ENABLES THE PRINTING OF THE LOCATION OF AN ERROR AND THE REGISTER
                                ;VALUES AT THE TIME OF THE ERROR.  THIS SHOULD BE SET TO NON-ZERO
                                ;ONLY WHEN DEBUGGING, OTHERWISE THROWS UP A LOT OF ''STRANGE'' NUMBERS
                                ;TO THE USER ON EVERY ERROR!


                                ;    THIS TELLS APL WHAT KIND OF PDP-11 IT WILL BE RUNNING ON, THE
                                ;DEFAULT IS A PDP-11/70


                                ;     'FPP' IS SET TO NON-ZERO IF A HARDWARE FPU IS AVAILABLE ON
                                ;THE PROCESSOR TO BE USED.  OTHERWISE IT SHOULD BE ZERO.


                                ;     'EIS' INDICATES WHETHER THE HARDWARE CONFIGURATION INCLUDES
                                ;THE EIS HARDWARE INSTRUCTION SET. IF NOT, SIMULATION SOFTWARE IS
                                ;GENERATED TO HANDLE THE INSTRUCTIONS.

000010                           FLTLEN=8.

                                ;    THIS IS THE NUMBER OF **BYTES** PER FLOATING POINT NUMBER.
                                ;THE ONLY POSSIBLE VALUES ARE 4 AND 10 .  WHEN SET TO 4, APL
                                ;IS ACCURATE TO 7 DIGITS IN MOST CASES.  WHEN SET TO 10, APL
                                ;IS ACCURATE TO 15 DIGITS IN MOST CASES.


                                ;    This Defines which Member of the RSX-Family (If any)
                                ; APL-11 Is Generated for use on.

000000                           RSTS=0

                                ;     When set to -1, it signifies that APL is going to be used
                                ; As a RSTS/E Run-Time System which means that we only generate
                                ; Pure Code.


                                .GLOBL  $MACHN, $FPP,   $FIS,   $RSX,   $RSTS,  $RT11,  $FPPLN

000106                                  $MACHN= MACHIN
000000                                  $EIS=   EIS
000000                                  $FIS=   FIS
000000                                  $FPP=   FPP
000115                                  $RSX=   RSX
000000                                  $RSTS=  RSTS
000000                                  $RT11=  RT11
000004                                  $FPPLN= FPPLEN
```

```
                        .SBTTL  DEFINITIONS     OPERATOR OVERLAYS

                        ; FOR RSX-11M, IAS, RT-11
                        ;       The functions and operators are placed in various overlays by
                        ;       the OVROP macro to save memory space.  To place a function in
                        ;       a specified overlay use OVROP with the derived function name
                        ;       and the overlay number.  Use 0 for the root, i.e. make it memory
                        ;       resident, and -1 to define a function without specifing an overlay.
                        ; FOR RSTS
                        ;       APL-11 V02.0 is configured as several RSTS Run-Time Systems. To
                        ;       specifiy which one(s) a module is in, a (1,2, or 4) is Specified.

                        .MACRO  OVROP   NAME, Z.RSX, Z.RT11, Z.RSTS
                            .IF NE RSX
                                TMPTAG  =Z.RSX                  ;RSX/IAS Organization
                            .IFF
                                .IF NE RT11
                                    TMPTAG  =Z.RT11             ;RT-11 Organization
                                .IFF
                                    TMPTAG  =Z.RSTS             ;RSTS/E Organization
                                .ENDC
                            .ENDC
                            .IRP NAM,<NAME>
                                NAM     =TMPTAG
                            .ENDR
                        .ENDM

                        ;                                       RSX  RT  RSTS
000000                  OVROP   <D.PLUS,D.$PLU>,                  0,  0,  3       ;ADDITION
000000                  OVROP   <D.SUB,D.$SUB>,                   0,  0,  3       ;SUBTRACTION
000000                  OVROP   <D.MUL,D.$MUL>,                   0,  0,  3       ;MULTIPLICATION
000000                  OVROP   D.DIV,                            0,  0,  3       ;DIVISION
000000                  OVROP   <D.MIN,D.$MIN>,                   0,  0,  3       ;MINIMUM
000000                  OVROP   <D.MAX,D.$MAX>,                   0,  0,  3       ;MAXIMUM
000000                  OVROP   <D.RES,D.$RES>,                   0,  0,  3       ;RESIDUE
000000                  OVROP   D.AND,                            0,  0,  3       ;AND
000000                  OVROP   D.OR,                             0,  0,  3       ;OR
000000                  OVROP   D.NN,                             0,  0,  3       ;NAND
000000                  OVROP   D.NR,                             0,  0,  3       ;NOR
000000                  OVROP   D.LT,                             0,  0,  3       ;LESS THAN
000000                  OVROP   D.LE,                             0,  0,  3       ;LESS THAN OR EQUAL
000000                  OVROP   D.EQ,                             0,  0,  3       ;EQUAL TO
000000                  OVROP   D.GE,                             0,  0,  3       ;GREATER THAN OR EQUAL
000000                  OVROP   D.GT,                             0,  0,  3       ;GREATER THAN
000000                  OVROP   D.NE,                             0,  0,  3       ;NOT EQUAL TO
000000                  OVROP   <M.IDN,M.$IDN>,                   0,  0,  1       ;IDENTITY
000000                  OVROP   <M.NEG,M.$NEG>,                   0,  0,  1       ;NEGATION
000000                  OVROP   <M.SIGN,M.$SIG>,                  0,  0,  1       ;SIGNUM
000000                  OVROP   M.RCIP,                           0,  0,  1       ;RECIPROCAL
000000                  OVROP   <M.$FLR,M.$CEI>,                  0,  0,  1       ;FLOATING FLOOR,  CEILING
000000                  OVROP   <M.FLR,M.CEI>,                   -1, -1, -1       ;INTEGER FLOOR,  CEILING
000000                  OVROP   <M.ABS,M.$ABS>,                   0,  0,  1       ;ABSOLUTE VALUE
000000                  OVROP   M.ROLL,                           0,  0,  1       ;ROLL
000000                  OVROP   M.PITM,                           0,  0,  1       ;PI TIMES
000000                  OVROP   M.NOT,                            0,  0,  1       ;NOT
000000                  OVROP   D.RHO,                            0,  0,  1       ;RESHAPE
000000                  OVROP   M.SHAP,                           0,  0,  1       ;SHAPE
```

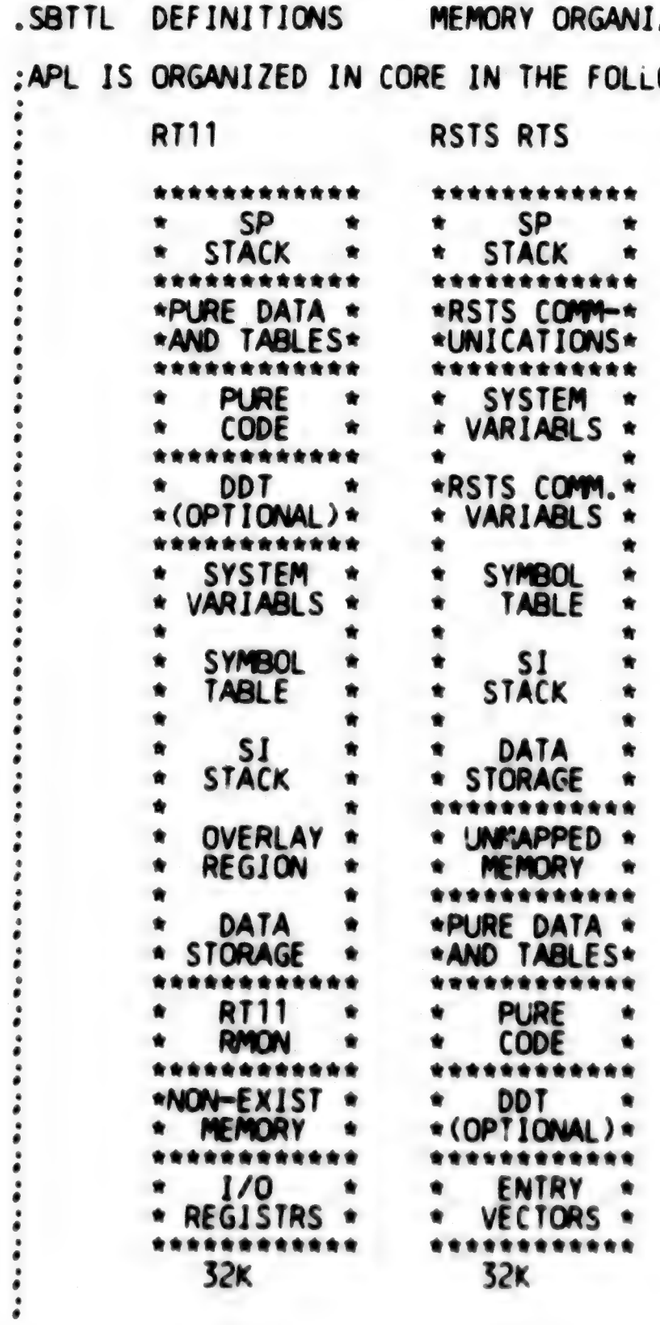
```
000000                  OVROP   D.CTLM,                 0,  0,  1       ;CATENATE / LAMINATE
000000                  OVROP   M.RAVL,                 0,  0,  1       ;RAVEL
000000                  OVROP   M.IOTA,                 0,  0,  1       ;INDEX GENERATOR
000000                  OVROP   M.EXE,                  0,  0,  2       ;EXECUTE
000000                  OVROP   <D.QUAD,M.QUAD>,        0,  0,  2       ;QUAD OUTPUT/INPUT
000000                  OVROP   <D.QQAD,M.QQAD>,        0,  0,  2       ;QUOTE QUAD OUTPUT/INPUT
000000                  OVROP   M.SUBR,                 1,  1,  2       ;SCIENTIFIC SUBROUTINES
000000                  OVROP   D.CIRC,                 1,  1,  3       ;TRIGOMETRIC FUNCTIONS
000000                  OVROP   D.PWR,                  1,  1,  3       ;POWER
000000                  OVROP   M.EXP,                  1,  1,  1       ;EXPONENTIAL
000000                  OVROP   D.LOG,                  1,  1,  3       ;LOGORITHUM
000000                  OVROP   M.LN,                   1,  1,  1       ;NATURAL LOGORITHUM
000000                  OVROP   D.COMB,                 1,  1,  3       ;COMBINATIONS
000000                  OVROP   M.FACT,                 1,  1,  1       ;FACTORIAL (GAMMA FUNCTION)
000000                  OVROP   M.SATAN,                1,  1,  1       ;USED BY SCAN,  REDUCTION,  MONADIC TRANS.
000000                  OVROP   <D.COMP,D.1COM>,        1,  1,  1       ;COMPRESSION
000000                  OVROP   D.ENCD,                 2,  2,  2       ;ENCODE
000000                  OVROP   D.DECD,                 2,  2,  2       ;DECODE
000000                  OVROP   D.IND,                  2,  2,  1       ;IOTA (INDEX OF)
000000                  OVROP   D.MEM,                  2,  2,  1       ;MEMBERSHIP
000000                  OVROP   <D.TAKE,D.DROP>,        2,  2,  1       ;TAKE / DROP
000000                  OVROP   <D.EXPD,D.1EXP>,        2,  2,  1       ;EXPANSION
000000                  OVROP   D.IBM,                  2,  2,  2       ;DYADIC I-BEAM
000000                  OVROP   <M.GDUP,M.GDWN>,        3,  3,  2       ;GRADE UP / DOWN
000000                  OVROP   <D.ROT,D.1ROT>,         3,  3,  2       ;ROTATE
000000                  OVROP   <M.REVS,M.1REV>,        3,  3,  2       ;REVERSE
000000                  OVROP   D.DEAL,                 3,  3,  1       ;DEAL
000000                  OVROP   M.IBM,                  3,  3,  2       ;MONADIC I-BEAM
000000                  OVROP   D.TRAN,                 4,  4,  2       ;DYADIC TRANSPOSE
000000                  OVROP   <D.FMT,M.FMT>,          4,  4,  1       ;FORMAT
000000                  OVROP   M.ENCD,                 4,  4,  2       ;QUOTE (MONADIC ENCODE)
000000                  OVROP   D.FPNT,                 5,  5,  2       ;FILE POINTER
000000                  OVROP   <D.FGET,D.FPUT>,        5,  5,  2       ;FILE INPUT / OUTPUT
000000                  OVROP   D.DQAD,                -1, -1, -1       ;DEL QUAD OUTPUT
000000                  OVROP   M.DQAD,                 6,  6,  2       ;DEL QUAD INPUT
000000                  OVROP   D.UNON,                 6,  6,  2       ;SET UNION
000000                  OVROP   D.INTR,                 6,  6,  2       ;SET INTERSECTION
000000                  OVROP   D.SDIF,                 6,  6,  2       ;SET DIFFERENCE
000000                  OVROP   <D.CONT,D.SCON>,        6,  6,  2       ;CONTAINS
000000                  OVROP   <D.SUBS,D.PSUB>,        6,  6,  2       ;SUBSET
000000                  OVROP   M.NUB,                  6,  6,  2       ;SELF INTERSECTION
000000                  OVROP   M.PRDSET,               6,  6,  2       ;USED BY INNER AND OUTER PRODUCT
000000                  OVROP   M.INSPAC,               6,  6,  2       ;GET SPACE FOR INNER PRODUCT
000000                  OVROP   M.DOUTER,               6,  6,  2       ;DO ACTUAL OUTER PRODUCT
000000                  OVROP   M.GETIDN,               6,  6,  1       ;USED BY REDUCTION, INNER PRODUCT
000000                  OVROP   <D.BXCT,M.BXCT>,        7,  7,  2       ;COMPARISON TOLERANCE
000000                  OVROP   <D.BXIO,M.BXIO>,        7,  7,  2       ;INDEX ORIGIN
000000                  OVROP   <D.BXPP,M.BXPP>,        7,  7,  2       ;PRINT PRECISION
000000                  OVROP   <D.BXPW,M.BXPW>,        7,  7,  2       ;PRINT WIDTH
000000                  OVROP   <D.BXRL,M.BXRL>,        7,  7,  2       ;RANDOM LINK
000000                  OVROP   M.BXAV,                 7,  7,  2       ;ATOMIC VECTOR
000000                  OVROP   M.BXCR,                 7,  7,  2       ;CANONICAL REPRESENTATION
000000                  OVROP   M.BXEX,                 7,  7,  2       ;Expunge
000000                  OVROP   M.BXLC,                 3,  3,  2       ;Line Counter
000000                  OVROP   M.BXFX,                 7,  7,  2       ;FIX FUNCTION
000000                  OVROP   M.BXNC,                 7,  7,  2       ;Name Classification
000000                  OVROP   <M.BXNL,D.BXNL>,        7,  7,  2       ;Name List
```

```
                                        OVROP   M.BXTS,                 7,  7,  2       ;TIME STAMP
                                        OVROP   M.BXWA,                 7,  7,  2       ;WORKSPACE AVAILABLE
                                        OVROP   <D.DMD,M.MMD>,          8., 8., 2       ;MATRIX DIVIDE / INVERSE
                                        OVROP   <M.RDCT,M.1RDT>,        9., 9., 1       ;REDUCTION
                                        OVROP   <M.SCAN,M.1SCN>,        9., 9., 1       ;SCAN
                                        OVROP   M.RAND,                10.,10., 2       ;RANDOM NUMBER GENERATOR
                                        OVROP   M.TRAN,                10.,10., 2       ;MONADIC TRANSPOSE
                                        OVROP   M.OUTR,                10.,10., 2       ;OUTER PRODUCT
                                        OVROP   D.INNR,                11.,11., 2       ;INNER PRODUCT

                                        ;IF 'OVROPR' IS DEFINED AS ZERO, APLOP ASSEMBLES RESIDENT CODE.  WHEN
                                        ;IT IS SET TO ONE BY APLOO, THE OVERLAY OPERATORS ARE ASSEMBLED.

                000000                  OVROPR= 0
```

```
                                .SBTTL  DEFINITIONS     REGISTERS

                                .ENABL AMA              ;FOR DEBUGGING!
                                .NLIST ME               ;BUT NOT EVERYTHING

                                ;GENERAL REGISTER DEFINITIONS

000000                                  R0=%0
000001                                  R1=%1
000002                                  R2=%2
000003                                  R3=%3
000004                                  R4=%4
000005                                  R5=%5
000006                                  R6=%6
000007                                  R7=%7

                                ; IN GENERAL, REGISTERS 5,6, AND 7 ARE DEDICATED TO
                                ; SPECIFIC PURPOSES AND MAY NOT BE 'TAMPERED'' WITH.
                                ; REGISTERS 2, 3, AND 4 USUALLY CONTAIN THE CURRENT TOKEN
                                ; BEING PARSED AND SHOULD BE PRESERVED ACROSS SUBROUTINES.
                                ; REGISTERS 0 AND 1 ARE SCRATCH REGISTERS AND NEED NOT
                                ; GENERALLY BE PRESERVED EXCEPT IN CERTAIN SPECIAL ROUTINES.
                                ; R0 IS MOST COMMONLY USED TO PASS PARAMETERS BETWEEN
                                ; ROUTINES.
                                ;
                                ; FLOATING POINT ACCUMULATORS
                                ;
000000                                  AC0=%0
000001                                  AC1=%1
000002                                  AC2=%2
000003                                  AC3=%3
000004                                  AC4=%4
000005                                  AC5=%5

                                ; FLOATING REGISTER USAGE IS NOT STANDARD EXCEPT FOR
                                ; SCALAR OPERATORS.  IN THIS CASE THE LEFT ARGUMENT
                                ; IS PASSED IN AC1, THE RIGHT ARG IN AC2, AND THE RESULT
                                ; IS RETURNED IN AC0.
                                ;
                                ; REGISTER 7, THE PROGRAM COUNTER, IS NOT GENERALLY
                                ; MODIFIED EXCEPT WHEN A CONSTANT IS ADDED TO
                                ; IT TO CAUSE A BRANCH THROUGH A TABLE.

000007                                  PC=%7

                                ; REGISTER 6 IS THE INTERPRETER STACK POINTER AND IS USED
                                ; MOSTLY FOR SUBROUTINE LINKAGE.

000006                                  SP=%6

                                ; REGISTER 5 IS DEDICATED TO THE STATE INDICATOR (SI) STACK
                                ; WHICH GROWS TOWARDS HIGH CORE.  R5 SHOULD ONLY BE MANIPULATED
                                ; WHEN 'INHIB' IS SET SO THAT RANDOM KEYBOARD INTERRUPTS DON'T
                                ; CAUSE STORAGE TO BE LOST OR DESTROYED IF R5 DOES NOT POINT
                                ; TO THE TOP OF AN SI FRAME.

000005                                  SI=%5
```

```
                                ; THE CONSTANT 'FISCON' IS USED TO CONVERT NUMBERS FROM INTEGER
                                ; TO FLOATING AND THE REVERSE.

046000                                  FISCON=46000
```

```
                        .SBTTL  DEFINITIONS     MEMORY ORGANIZATION

                        ;APL IS ORGANIZED IN CORE IN THE FOLLOWING MANNER:
```



```
                                .SBTTL  DEFINITIONS     ORDERING PSECTS

                                ;-----------------------------------------------
                                ;               TABLES AND PURE DATA
000000                          .PSECT  TABLES, RO, D, LCL      ;Interpreter Tables (Pure)
000000                          .PSECT  MONTB,  RO, D, LCL      ;Monadic Operator Branch Table
000000                          .PSECT  DYDTB,  RO, D, LCL      ;Dyadic  Operator Branch Tab;e
000000                          .PSECT  BOPTB,  RO, D, LCL      ;Operator Description Bits Table
000000                          .PSECT  MONBTB, RO, D, LCL      ;Monadic Operator Bits
000000                          .PSECT  DYDBTB, RO, D, LCL      ;Dyadic  Operator Bits
000000                          .PSECT  COMTB,  RO, D, LCL      ;Command Tables
000000                          .PSECT  CHRTBL, RO, D, LCL      ;APL Character Description Bits
000000                          .PSECT  VTBL,   RO, D, LCL      ;Speciality Values of APL Characters
000000                          .PSECT  ASCTBL, RO, D, LCL      ;ASCII Equivalents of APL Characters
000000                          .PSECT  OVRTBL, RO, D, LCL      ;Table of Overstruck  APL Characters
000000                          .PSECT  TRNTBL, RO, D, LCL      ;APL Equivalents of ASCII Characters
000000                          .PSECT  MNTBL,  RO, D, LCL      ;ASCII Mneumonics Table
000000                          .PSECT  MNTRTL, RO, D, LCL      ;APL Value of ASCII Mneumonic
                                ;               TABLES and PURE DATA
                                ;-----------------------------------------------
                                ;               PURE CODE
000000                          .PSECT  CODE,   RO, I, LCL      ;Interpreter Code (Pure)
000000                          .PSECT  PATCH,  RO, I, GBL      ;APL Debugging/Patch Area
                                ;               PURE CODE
                                ;-----------------------------------------------
                                ;               IMPURE
```

```
                        .SBTTL  DEFINITIONS     SI PARSE FRAMES

                        ; DURING THE EVALUATION OF EXPRESSIONS, THE STATE
                        ; INDICATOR STACK CONTAINS THE CURRENT STATE OF THE
                        ; PARSE (NATURALLY!)  THE SI STACK GROWS UP IN FREE
                        ; CORE FROM THE END OF THE SYMBOL TABLE, WHILE USER
                        ; STORAGE GROWS DOWN TOWARDS IT FROM HIGH CORE.
                        ;
                        ; STACK FRAME MAP:
                        ;
                        ;                       SI.LEN
                        ;           ************
                        ;           *  RETURN  *    SI.RET
                        ;           ************
                        ;           *   BITS   *    SI.BIT
                        ;           ************
                        ;           *  VALUE   *    SI.VAL
                        ;           ************
                        ;           *  NEXT    *    <-- REFERENCE POINT
                        ;           *     FRAME*
                        ;
                        ;
 177772                         SI.LEN=-6          ;LENGTH OF SI FRAME
 177772                         SI.RET=-6          ;SUBROUTINE RETURN POINTER
 177774                         SI.BIT=-4          ;TOKEN DESCRIPTION BITS
 177776                         SI.VAL=-2          ;VALUE OF TOKEN


                        ; A FRAME IS ON ''TOP'' OF THE SI STACK WHEN THE SI
                        ; POINTER POINTS TO THE WORD PAST THE FRAME.
                        ;
                        ;SI.RET CONTAINS THE SUBROUTINE RETURN WHICH IS
                        ;       IS TO BE TAKEN WHEN THE FRAME ON TOP OF THE
                        ;       STACK IS ASSIGNED A VALUE.  THIS ALLOWS
                        ;       RECURSION WITHOUT FILLING UP THE INTERPRETER
                        ;       STACK.
                        ;
                        ;SI.BIT CONTAINS BITS DESCRIBING THE VALUE IN THE SI
                        ;       FRAME.  SEE THE EXPLANATION OF TOKEN DESCRIPTION
                        ;       BITS BELOW.  A ZERO WORD INDICATES THAT NO
                        ;       VALUE WAS ASSIGNED TO THE FRAME WHEN IT WAS
                        ;       EVALUATED.
                        ;
                        ;SI.VAL IS THE VALUE OF THE SI FRAME, AND IS
                        ;       DESCRIBED BY SI.BIT.  USUALLY IT IS A POINTER
                        ;       TO A BLOCK OF STORAGE, AN OPERATOR, OR A
                        ;       SPECIAL CHARACTER SUCH AS A BRACKET.
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 10
DEFINITIONS     TOKEN DESCRIPTION BITS


                                .SBTTL  DEFINITIONS     TOKEN DESCRIPTION BITS

                                ; THE BITS BELOW DESCRIBE THE VALUE OF A STATE
                                ; INDICATOR STACK FRAME OR TOKEN.
                                ;
000001                          T.OPFN  =000001 ;VALUE IS AN OPERATOR OR A FUNCTION
000002                          T.SOP   =000002 ;OPERATOR IS A SCALAR OP
000004                          T.NIL   =000004 ;OP IS NILADIC (NO ARGUMENTS)
000010                          T.MON   =000010 ;OP IS MONADIC (ELSE OP IS DYADIC)
000020                          T.LIT   =000020 ;OP TAKES LITERAL ARGUMENTS
000040                          T.SPOP  =000040 ;OP IS SPECIAL OP (MAY BE ASSIGNED
                                                ;A VALUE SUCH AS A QUAD)
000100                          T.DIM   =000100 ;OP TAKES OPTIONAL DIMENSION ARGUMENT
000200                          T.TMP   =000200 ;VALUE IS TEMPORARY BLOCK, JUST DELETE IT WITH KILL
000400                          T.LIST  =000400 ;VALUE IS POINTER TO LIST BLOCK
001000                          T.VAR   =001000 ;VALUE IS SYMBOL TABLE POINTER
002000                          T.NUM   =002000 ;VALUE IS NUMERIC (ELSE IT IS LITERAL)
004000                          T.CON   =004000 ;VALUE IS POINTER TO CODESTRING CONSTANT
010000                          T.FUN   =010000 ;VALUE IS POINTER TO FUNCTION'S NAME.
                                                ; ELSE VALUE IS OPERATOR NUMBER
020000                          T.BOL   =020000 ;VALUE IS BEGINNING OR END OF LINE
040000                          T.OTHR  =040000 ;VALUE IS SINGLE CHARACTER
100000                          T.VAL   =100000 ;VALUE IS POINTER TO ARRAY IN STORAGE
                                ;
                                ; NOTE THAT SOME OF THESE BITS MAY BE USED TOGETHER
                                ; SUCH AS:  T.OPFN+T.FUN+T.MON
                                ;
                                ;ALSO  ALL BITS AFFECTING OPERATORS MUST FIT IN
                                ;THE LOWER BYTE BECAUSE OF 'BOPTAB'!!!!!!!!!!!!
                                ;
                                ;
                                ;
                                ; THESE BITS GO INTO DYDBTL and MONBTL, and specify
                                ; data about the scalar operators and who they take as arguments
                                ;
000010                          CS.BOOL =000010 ;RESULT OF OP WITH BOOLEAN SOURCES IS BOOLEAN
000020                          CS.FLOAT=000020 ;RESULT OF OP WITH FLOAT SOURCES WILL ATTEMPT BOOLEAN
000040                          CS.IDN  =000040 ;OPERATION HAS IDENTITY ELEMENT
000100                          CS.ONE  =000100 ;OPERATION IDENTITY ELEMENT IS 1 (ELSE 0)
000200                          CS.ASS  =000200 ;OPERATION IS ASSOCIATIVE  (FOR SCAN)
```

```
                        .SBTTL  DEFINITIONS     SI EXECUTION FRAMES

                        ;AN EXECUTION FRAME (E-FRAME) INSTRUCTS APL TO
                        ;READ AND EVALUATE A LINE ENTERED FROM THE CURRENT
                        ;INPUT DEVICE.  AN E FRAME IS SPAWNED ON THE SI
                        ;STACK WHEN APL IS INITIALIZED, WHEN A FUNCTION IS
                        ;SUSPENDED, AND WHEN A REQUEST IS MADE FOR EVALUATED
                        ;INPUT.  'SIFRAME' POINTS TO THE FRAME CURRENTLY ON
                        ;THE TOP OF THE SI STACK.
                        ;
                        ;       ********************
                        ;       *      RETURN      *      F.RET
                        ;       ********************
                        ;       *       BITS       *      F.BITS
                        ;       ********************
                        ;       *      VALUE       *      F.VAL
                        ;       ********************
                        ;       * PREVIOUS FRAME   *      F.PREV  <-- REFERENCE POINT
                        ;       ********************
                        ;       *      FLAGS       *      F.FLAG
                        ;       ********************
                        ;       *  LAST C-BYTE     *      F.LSTC
                        ;       ********************
                        ;       *  HEAD POINTER    *      F.HEAD
                        ;       ********************
                        ;       *                  *      F.CODE
                        ;       *                  *
                        ;       *   CODESTRING     *
                        ;       *                  *
                        ;       ********************
                        ;       *                  *
                        ;       *  PARSE FRAMES    *
                        ;       *                  *


000000                  F.RET=0         ;CURRENT TOKEN POINTER WHEN SUSPENDED
000002                  F.BITS=2        ;CURRENT TOKEN BITS WHEN SUSPENDED
000004                  F.VAL=4         ;CURRENT TOKEN VALUE WHEN SUSPENDED
000006                  F.PREV=6        ;POINTER TO PREVIOUS FRAME OR ZERO
000010                  F.FLAG=10       ;FLAG BITS DESCRIBING FRAME
000012                  F.LSTC=12       ;POINTER TO LAST CODESTRING BYTE
000014                  F.HEAD=14       ;POINTER PAST HEADER DATA IN FRAME
000016                  F.CODE=16       ;OFFSET IN WHICH FIRST BYTE OF CODESTRING
                                        ;WILL BE PLACED BY PASS 2 (LEX)

                        ;BELOW ARE THE BIT ASSIGNMENTS FOR THE FLAGS (F.FLAG)

000002                  F.SUSP= 000002  ;F-FRAME IS SUSPENDED
100000                  F.PRNT= 100000  ;PRINT THE RESULTS OF THIS FRAME
000004                  F.STOP= 000004  ;A CLOSING DEL HAS BEEN FOUND DURING FN DEF
000010                  F.FNDF= 000010  ;IN FUNCTION DEFINITION MODE
000020                  F.BR=   000020  ;BRANCH FOUND IN FUNCTION EXECUTION
000040                  F.REDF= 000040  ;FUNCTION HEADER IS BEING RE-DEFINED
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 11-1
DEFINITIONS     SI EXECUTION FRAMES

     572            000100                  F$EVAL= 000100  ;FRAME IS KEYBOARD EVALUATED INPUT FRAME
     573            000200                  F$EVI=  000200  ;FRAME IS EVALUATED INPUT FRAME (QUAD)
     574            000400                  F$EXEC= 000400  ;FRAME IS EXECUTION FRAME (EXECUTE OPERATOR)
     575            001000                  F$FUNC= 001000  ;FRAME IS F-FRAME (FUNCTION CALL)
     576            002000                  F.END=  002000  ;END OF EXECUTE STRING HAS BEEN REACHED
     577            004000                  F.READ= 004000  ;USER IS READING A FILE
     578            010000                  F.PROT= 010000  ;FUNCTION WILL BE LOCKED
```

```
                                .SBTTL  DEFINITIONS     SPECIAL PARSE CHARACTERS

                                ; THE FOLLOWING SPECIAL CHARACTERS APPEAR; IN A
                                ; CODESTRING TO MARK SPECIAL DATA OR FOR OTHER
                                ; PURPOSES.  SOME MARKERS MUST FALL ON EVEN OR
                                ; ODD BYTE BOUNDARIES BECAUSE THEY MUST FOLLOW
                                ; WORDS OF DATA.  FILL CHARACTERS ARE ADDED WHEN
                                ; NECESSARY.  REMEMBER THAT CODESTRINGS ARE
                                ; EVALUATED FROM RIGHT TO LEFT (HIGH TO LOW
                                ; ADDRESSES)
                                ;

000177                          M.FIL   =177    ;FILL CHARACTER, TO BE IGNORED
000200                          M.BOL   =200    ;BEGINNING OF LINE
000201                          M.SYN   =201    ;CAUSES SYNTAX ERROR
000202                          M.CHR   =202    ;CAUSES CHARACTER ERROR
000203                          M.SYM   =203    ;FOLLOWS WORD POINTING INTO SYMBOL TABLE
000204                          M.NUM   =204    ;INDICATES NUMERIC CONSTANT.  PRECEEDING
                                                ;BYTE CONTAINS COUNT OF FLOATING POINT
                                                ;VECTOR ELEMENTS IN PRECEEDING WORDS.
000205                          M.LIT   =205    ;INDICATES LITERAL CONSTANT. PRECEEDING
                                                ;BYTE CONTAINS COUNT OF CHARACTERS IN
                                                ;CONSTANT.
000206                          M.CMT   =206    ;INDICATES N BYTES TO BE SKIPPED OVER.
                                                ;PREVIOUS BYTE CONTAINS N.
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 13
DEFINITIONS     CHRTAB BITS

                                .SBTTL  DEFINITIONS     CHRTAB BITS

                                ; THESE BITS ARE USED TO DESCRIBE APL CHARACTERS.
                                ;

000001                          C.SPLX  =001    ;CHAR IS SPECIAL LEX CHARACTER
000002                          C.SPRS  =002    ;CHAR IS SPECIAL PARSE CHARACTER
000004                          C.OP    =004    ;CHAR IS OP
000010                          C.IDEN  =010    ;CHAR IS PART OF IDENTIFIER (A-Z, DELTA)
000020                          C.NUM   =020    ;CHAR IS NUMERIC (0-9,-,.)
000040                          C.DIG   =040    ;CHAR IS A DIGIT (0-9)
000100                          C.SPFX  =100    ;CHAR IS SPECIAL FIXLIN CHAR
000200                          C.SPGL  =200    ;CHAR IS SPECIAL TO 'GETL'



                                .SBTTL  DEFINITIONS     SYSTEM STATUS BITS

                                ;THESE BITS APPEAR IN 'SYSTAT' AS TO THE CONDITION OF THE APL SYSTEM

000001                          SY.BAD= 000001          ;POSSIBLE WORK SPACE DAMAGE
000002                          SY.OPN= 000002          ;CLEAR FILE OPEN ON CHANNEL 15. (RTS VERSION)
000004                          SY.SVC= 000004          ;TTY INPUT VECTORS SAVED (RT11 VERSION)
000010                          SY.BS=  000010          ;ALLOW BACKSPACES ON RUBOUTS
000020                          SY.ERP= 000020          ;DON'T PRINT ON EPSILON ERRORS
000040                          SY.CLR= 000040          ;CLEARING SI STACK
000100                          SY.NOU= 000100          ;NO OUTPUT ALLOWED (LOAD COMMAND)
000200                          SY.SVL= 000200          ;SPECIAL SAVE/LOAD I/O MODE
000400                          SY.SRT= 000400          ;STARTING UP, CHECK FOR ''CONTINUE'' WORKSPACE
001000                          SY.PMT= 001000          ;PROMPT FOR READ IS REALLY SPACES
100000                          SY.BRK= 100000          ;BREAK CHARACTER ENCOUNTERED
```

```
                        .SBTTL  DEFINITIONS     F-FRAMES

                        ;AN F-FRAME (FUNCTION EVALUATION FRAME) IS SPAWNED ON THE
                        ;SI STACK BY 'FNCALL' WHEN A USER DEFINED FUNCTION IS CALLED.
                        ;THE FIRST PART OF AN F-FRAME LOOKS LIKE AN E-FRAME, EXCEPT
                        ;THAT 'F.LSTC' POINTS PAST THE LAST LABEL VALUE IN THE FRAME
                        ;(SEE BELOW.)  THE REST OF THE F-FRAME CONTAINS A POINTER
                        ;TO THE HEADER BLOCK FOR THE FUNCTION, A POINTER TO THE CURRENT
                        ;LINE BLOCK BEING EXECUTED, AND A LIST OF THE OLD VALUES
                        ;OF THE LOCAL VARIABLES.
                        ;  WHEN A FUNCTION IS CALLED, THE VALUES OF THE LOCAL SYMBOLS
                        ;('S.VAL' IN THE SYMBOL BLOCKS) ARE PUSHED ON THE SI
                        ;STACK SO THAT WHEN THE FUNCTION EXITS ('FNEXIT'), THE VALUES
                        ;OF THE LOCAL VARIABLES CAN BE RESTORED SO THAT IT LOOKS
                        ;LIKE NOTHING HAPPENED.  WHEN A FUNCTION IS EXITED, THE RESULT
                        ;IF ANY IS PLACED IN A PARSE FRAME ON THE SI STACK AND A RETURN
                        ;IS MADE TO THE FUNCTION CALLER.
                        ;
                        ;DESCRIPTION OF AN F-FRAME:
                        ;
                        ;       ********************
                        ;       *      RETURN      *      SI.RET
                        ;       ********************
                        ;       *       BITS       *      SI.BIT
                        ;       ********************
                        ;       *      VALUE       *      SI.VAL
                        ;       ********************
                        ;       * PREVIOUS FRAME  *      F.PREV  <-- REF. PT.
                        ;       ********************
                        ;       *      FLAGS       *      F.FLAG
                        ;       ********************
                        ;       *   LAST LOCAL     *      F.LSTC
                        ;       ********************
                        ;       *  HEAD POINTER    *      F.HEAD
                        ;       ********************
                        ;       *FUNCTION HEADER   *      F.FUN
                        ;       ********************
                        ;       *  LINE POINTER    *      F.LINE
                        ;       ********************
                        ;       *  INPUT POINTER   *      F.R2
                        ;       ********************
                        ;       *   TOKEN BITS     *      F.R3
                        ;       ********************
                        ;       *  TOKEN VALUE     *      F.R4
                        ;       ********************
                        ;       *      RETURN      *      F.STK
                        ;       ********************
                        ;       *      RESULT      *
                        ;       ********************
                        ;       *     LEFT ARG     *
                        ;       ********************
                        ;       *     RIGHT ARG    *
                        ;       ********************
                        ;       *     LOCAL 1      *
                        ;       ********************
                        ;       *                  *
                        ;
```

```
                                ;       *               *
                                ;
                                ;       *    LOCAL N    *
                                ;       ****************
                                ;       *    LABEL 1    *
                                ;       ****************
                                ;       *               *
                                ;
                                ;       *               *
                                ;
                                ;       *               *
                                ;       ****************
                                ;       *    LABEL N    *
                                ;       ****************
                                ;
                                ;F.PREV=0       ;PTR TO PREVIOUS SI FRAME
                                ;F.FLAG=2       ;FLAGS DESCRIBING FRAME
                                ;F.LSTC=4       ;POINTER PAST LAST LABEL IN FRAME
                                ;F.HEAD=6       ;POINTER PAST HEADER DATA
000010                          F.FUN=10        ;PTR TO FUNCTION HEADER
000012                          F.LINE=12       ;PTR TO CURRENT LINE BLOCK
000014                          F.R2=14         ;OLD INPUT CODESTRING POINTER
000016                          F.R3=16         ;OLD TOKEN BITS
000020                          F.R4=20         ;OLD TOKEN VALUE
000022                          F.STK=22        ;TOP ITEM ON SP STACK (CALLER RETURN)
```

```
                        .SBTTL  DEFINITIONS     FUNCTION LINE BLOCKS

                        ;USER DEFINED FUNCTIONS CONSIST OF A CIRCULAR LIST OF FUNCTION
                        ;LINE BLOCKS.  LINE #0 OF A FUNCTION IS THE FUNCTION HEADER,
                        ;AND IT CONTAINS POINTERS TO THE ARGUMENT SYMBLS, LOCAL VARIABLES,
                        ;PLUS A POINTER BACK TO THE NAME BLOCK IN THE SYMBOL TABLE FOR
                        ;THE FUNCTION.  THE FUNCTION HEADER IS A SPECIAL CASE OF THE
                        ;LINE BLOCK (SEE BELOW)  EACH LINE BLOCK CONTAINS THE CODESTRING
                        ;WHICH MAKES UP THE FUNCTION LINES, THE LINE NUMBER, AND VARIOUS
                        ;OTHER JUNK NECESSARY FOR FUNCTION EXECUTION AND DEFINITION.
                        ;
                        ;FORMAT OF A FUNCTION LINE BLOCK:
                        ;FUNCTION HEADER DEFINITIONS ARE IN PARENTHESES)
                        ;
                        ;       ******************
                        ;       *     LENGTH     *
                        ;       ******************
                        ;       *CODESTRING OFF.*       R.OFF (R.PTR) <-- REF. PT.
                        ;       ******************
                        ;       * BITS * LINE # *       R.BITS / R.LN (R.LCT)
                        ;       ******************
                        ;       * PREVIOUS LINE *       R.PREV
                        ;       ******************
                        ;       *   NEXT LINE    *       R.NEXT
                        ;       ******************
                        ;       *     LINE       *       R.LNUM
                        ;       *    NUMBER      *
                        ;       ******************
                        ;       *  LABEL NAME    *
                        ;       ******************
                        ;       *    (RESULT)    *
                        ;       ******************
                        ;       *  (LEFT ARG)    *
                        ;       ******************
                        ;       *  (RIGHT ARG)   *
                        ;       ******************
                        ;       *    (LOCAL 1)   *
                        ;       ******************
                        ;       *                *
                        ;
                        ;       *  CODESTRING    *
                        ;
                        ;       *                *
                        ;       ******************
                        ;       *     LENGTH     *
                        ;       ******************
```

```
                                
000000                  R.OFF=0         ;OFFSET TO END OF CODESTRING
000000                  R.PTR=0         ;POINTER TO FN'S NAME SYMBOL BLOCK
000002                  R.LN=2          ;BYTE CONTAINING LINE #
000002                  R.LCT=2         ;(COUNT OF LOCAL VARIABLES IN FN)
000003                  R.BITS=3        ;BITS DESCRIBING LINE (FUNCTION)
000004                  R.PREV=4        ;POINTER TO PREVIOUS (OR LAST) LINE BLOCK
000006                  R.NEXT=6        ;POINTER TO NEXT (OR FIRST) LINE BLOCK
000010                  R.LNUM=10       ;FLOATING LINE # (0=FUNCTION HEADER)

                        ;THE LENGTH OF THE LINE NUMBER (IN BYTES) IS DETERMINED BY
                        ;THE FLOATING POINT MODE (SINGLE OR DOUBLE).  IF A LABEL
                        ;IS FOUND ON THE LINE, A POINTER TO IT'S SYMBOL TABLE
                        ;BLOCK FOLLOWS THE LINE NUMBER.  THEN COMES A FAKE ARRAY
                        ;BLOCK WHICH POINTS TO THE LINE NUMBER AS THE VALUE OF THE
                        ;ARRAY.  THIS IS WHY THE LINE NUMBER OF THE FUNCTION IS
                        ;STORED AS A FLOATING POINT NUMBER.  THUS, KILLING A FUNCTION
                        ;LINE CONSISTS SIMPLY OF DEALLOCATING IT.  THIS WILL KILL
                        ;THE ARRAY BLOCK FOR THE OPTIONAL LABEL AT THE SAME TIME,
                        ;WHICH GREATLY SPEEDS THINGS UP.
                        ;
                        ;VALUES FOR BITS BYTE:
                        ;
000100                  F.FLB=100       ;THIS IS A FUNCTION LINE BLOCK
000040                  F.RES=040       ;FUNCTION RETURNS A RESULT
000020                  F.LBL=020       ;FUNCTION LINE HAS A LABEL
000010                  F.OPN=010       ;FUNCTION DEFINITION IS CURRENTLY OPEN
000004                  F.LOC=004       ;FUNCTION IS LOCKED
000002                  F.TRC=002       ;FUNCTION LINE HAS A TRACE ON IT
000001                  F.STP=001       ;FUNCTION LINE HAS A STOP ON IT

000002                  F.MON=2         ;FUNCTION IS MONADIC (ELSE DYADIC)
000001                  F.NIL=1         ;FUNCTION IS NILADIC

040000                  S.FLB=0'0000    ;SPECIFY FUNCTION LINE BLOCK
```

```
                                .SBTTL  DEFINITIONS     ARRAY STORAGE BLOCK

                                ;ARRAYS (IN FACT ALL NUMERIC AND LITERAL VARIABLES) ARE STORED
                                ;IN BLOCKS OF THE FOLLOWING FORM:

                                ;       **************
                                ;       *  LENGTH    *
                                ;       **************
                                ;       *  OFFSET    *   A.PTR   <-- REFERENCE POINT
                                ;       **************
                                ;       *BITS*# DIMS*   A.BITS  /  A.DIMS
                                ;       **************
                                ;       *   ALEN     *   A.ALEN
                                ;       **************
                                ;       *   DIM1     *   A.DIM1
                                ;       *            *
                                ;       *    ...     *
                                ;       **************
                                ;       *   VALUE    *
                                ;       *            *
                                ;       **************
                                ;       *  LENGTH    *
                                ;       **************


000000                          A.PTR=0         ;OFFSET TO FIRST ELEMENT IN ARRAY
000002                          A.DIMS=2        ;NUMBER OF DIMENSIONS IN ARRAY
000003                          A.BITS=3        ;BITS DESCRIBING ARRAY
000004                          A.ALEN=4        ;VOLUME OF ARRAY (TOTAL # OF ELEMENTS)
000006                          A.DIM1=6        ;MAXIMUM VALUE OF LAST INDEX.  FOLLOWING
                                                ;A.DIM1 IS A LIST OF ALL OF THE MAXIMUM
                                                ;INDICES OF THE ARRAY. A.DIMS CONTAINS
                                                ;THE COUNT OF THE ELEMENTS IN THIS LIST.
000020                          DIMAX=16.       ;MAXIMUM NUMBER OF DIMENSIONS ALLOWED


                                ;NEXT COME THE ELEMENTS IN THE ARRAY IN ROW-MAJOR ORDER.
                                ;THE OFFSET OF THE FIRST ELEMENT OF THE ARRAY FROM A.PTR
                                ;IS CONTAINED IN A.PTR.
                                ;
                                ;THE LENGTH WORDS AT EACH END OF THE BLOCK ARE USED FOR
                                ;BOOKKEEPING BY THE STORAGE MANAGEMENT ROUTINES.
                                ;
                                ;       BLOCK DESCRIPTION BITS:
                                ;
000006                          A.NUM=006       ;VALUE IS NUMERIC (ELSE LITERAL)
000010                          A.TMP=010       ;VALUE IS NOT TEMPORARY
000020                          A.CON=020       ;VALUE IS CODESTRING CONSTANT
000040                          A.ARG=040       ;VALUE IS COPIED ARGUMENT FOR USER-DEFINED FUNCTION
000200                          A.ASB=200       ;THIS IS AN ARRAY STORAGE BLOCK
                                ;
                                ;       DATA TYPES:
                                ;
000000                          .CHARACTER=     0
000002                          .BOOLEAN=       2
000004                          .INTEGER=       4
```

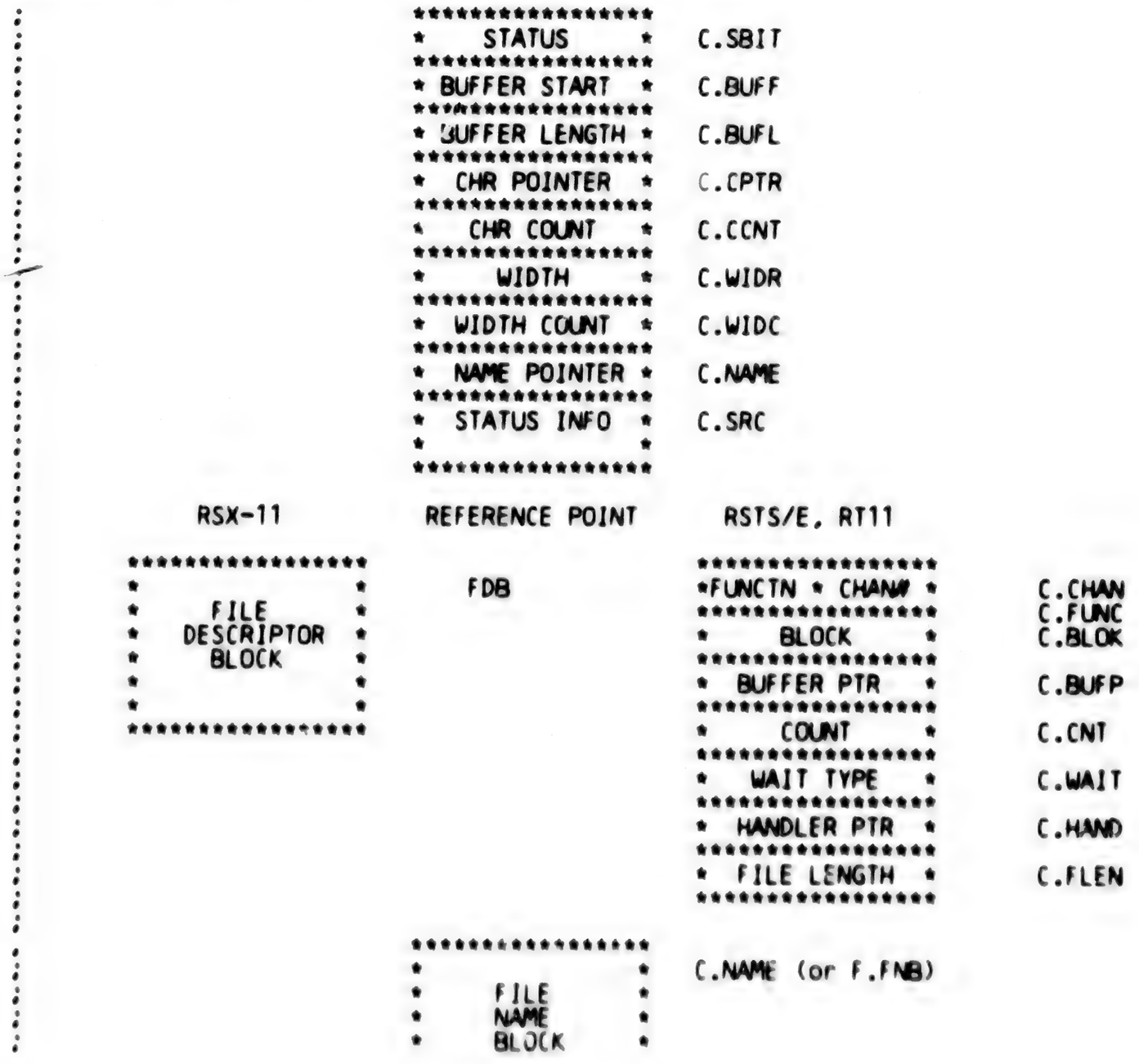
```
     864          000006                  .FLOAT=         6
     865
     866          140000                  S.TYPE  =140000 ;MASK FOR DETERMINING TYPE
     867          100000                  S.ASB   =100000 ;SPECIFY ARRAY STORAGE BLOCK
```

```
                                .SBTTL  DEFINITIONS     LIST STORAGE BLOCK

                                ;A LIST BLOCK IS USUALLY CREATED DURING SUBSCRIPT
                                ;CALCULATIONS.  EACH ENTRY IN THE LIST BLOCK IS A
                                ;POINTER TO A STORAGE BLOCK.  THE STORAGE BLOCKS
                                ;REPRESENT THE ARRAYS WHICH COMPRISE THE SUBSCRIPTS
                                ;THUS, EACH APL EXPRESSION DELIMITED BY SEMI-COLONS
                                ;IS POINTED TO BY THE POINTERS IN A LIST BLOCK.  THE
                                ;POINTERS ARE ADDED WORKING BACKWARDS FROM THE END
                                ;OF THE LIST BLOCK TOWARDS ITS HEAD.  IF THE BLOCK
                                ;OVERFLOWS, A LARGER BLOCK IS ALLOCATED, AND THE
                                ;POINTERS ARE COPPIED INTO THE NEW BLOCK.  (THIS
                                ;RARELY HAPPENS).
                                ;
                                ;       *************
                                ;       *  LENGTH   *
                                ;       *************
                                ;       * POINTER  *    L.PTR   <-- REFERENCE POINT
                                ;       *************
                                ;       *BITS*COUNT*    L.BITS  /  L.CNT
                                ;       *************
                                ;       *LAST PTR  *    L.PTR1
                                ;       *          *
                                ;       * POINTERS *
                                ;       *          *
                                ;       *          *    FIRST POINTER
                                ;       *************
                                ;       *  LENGTH   *
                                ;       *************
                                ;
000000                          L.PTR=0         ;POINTS TO LAST ENTRY STORED IN LIST BLOCK
000002                          L.CNT=2         ;NEGATIVE COUNT OF WORDS AVAILABLE IN BLOCK
000003                          L.BITS=3        ;BITS DESCRIBING LIST
000004                          L.PTR1=4        ;OFFSET TO LAST AVAILABLE POINTER

000020                          LISIZE=20       ;INITIAL SIZE OF BLOCK

000001                          L.PRNT=001      ;PRINTING FLAG TEMP SAVE
000002                          L.SUBS=002      ;Used for Subscripting (KILL must delete subsidiaries)
```

```
                                .SBTTL  DEFINITIONS     INDEXING STACK FRAME

                                ;A LIST OF INDEXING STACK FRAMES IS CREATED ON THE
                                ;SP STACK DURING SUBSCRIPT CALCULATIONS.  ONE INDEX
                                ;FRAME IS CREATED FOR EACH DIMENSION IN THE ARRAY
                                ;BEING SUBSCRIPTED.  BELOW IS A MAP OF AN INDEXING
                                ;FRAME:
                                ;
                                ;       *********
                                ;       * STRIDE*        I.STR
                                ;       *********
                                ;       *  OLD  *        I.OLD
                                ;       *********
                                ;       *MAX DIM*        I.DIM
                                ;       *********
                                ;       *POINTER*        I.PTR
                                ;       *********
                                ;       * COUNT *        I.CNT
                                ;       *********
                                ;       *       *        <-- REFERENCE POINT
                                ;
177766                          I.STR=-12       ;STRIDE FOR THIS DIMENSION
177770                          I.OLD=-10       ;VALUE OF LAST INDEX IN THIS DIMENSION
177772                          I.DIM=-6        ;MAXIMUM INDEX FOR THIS DIM
177774                          I.PTR=-4        ;POINTS TO VALUES IN ARRAY COMPRISING
                                                ;THE LIST OF INDICES FOR THIS DIMENSION
177776                          I.CNT=-2        ;COUNT OF INDICES POINTED TO BY I.PTR
000012                          I.LEN=12        ;OFFSET TO BACK UP BY 1 INDEX FRAME
```

```
                                .SBTTL  DEFINITIONS     SYMBOL TABLE ENTRY

                                ;AN ENTRY IN THE SYMBOL TABLE HAS THE FOLLOWING FORM:
                                ;
                                ;       ************
                                ;       *  LINK    *      S.LINK
                                ;       ************
                                ;       * ABLINK  *      S.AB
                                ;       ************
                                ;       *   VAL    *      S.VAL   <-- REFERENCE POINT
                                ;       ************
                                ;       *CHR1*LEN *      S.CHR1  /  S.LEN
                                ;       ************
                                ;       * REST OF *
                                ;       *  ENTRY  *
                                ;       ************

177774                          S.LINK=-4       ;LINK TO NEXT ENTRY.  EACH WORD IN THE
                                                ;HASH TABLE POINTS TO A LINKED LIST OF
                                                ;ENTRIES WHICH HASH TO THAT LOCATION.
177776                          S.AB=-2         ;ALL ENTRIES ARE LINKED TOGETHER IN
                                                ;ALPHABETICAL ORDER TO MAKE LISTING EASY.
000000                          S.VAL=0         ;POINTS OF VALUE OF SYMBOL. (A BLOCK OF
                                                ;STORAGE) OR IS ZERO IF NO VALUE.
000002                          S.BITS=2        ;BITS DESCRIBING SYMBOL.
000002                          S.LEN=2         ;NUMBER OF CHARACTERS IN SYMBOL
000003                          S.CHR1=3        ;FIRST CHAR OF SYMBOL
```

```
                        .SBTTL  DEFINITIONS     CHANNEL BLOCK

                        ;ALL I/O IS DONE THROUGH THESE BLOCKS.  CHANNEL 0 IS RESERVED FOR THE
                        ;USER'S TERMINAL I/O.  CHANNEL 15 IS USED FOR OVERLAYS AND NON-RESIDENT
                        ;CODE ON RT-11/RSTS.  CHANNEL 14 IS RESERVED FOR A SYSTEM UTILTY
                        ;CHANNEL.  CHANNEL 13 IS RESERVED FOR FUTURE USE OR SPECIAL USER
                        ;APPLICATIONS.  THIS LEAVES CHANNELS 1 THROUGH 12 AVAILABLE FOR USER
                        ;MANIPULATION.  DUE TO THE NATURE OF I/O, CHANNEL BLOCKS ARE NOT PART OF
                        ;THE STANDARD DATA AREA, BUT ARE PERMANENT BLOCKS.

000020                  NUMCHN= 16.                     ;NUMBER OF CHANNELS
000014                  MAXCHN= 12.                     ;MAXIMUM USER CHANNEL
```



```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 21-1
DEFINITIONS     CHANNEL BLOCK

    1026                                :                   *                 *
    1027                                :                   *****************
```

DEFINITIONS CHANNEL BLOCK

```
1029
1030          177756          C.SBIT= -22             ;STATUS BITS FOR CHANNEL
1031          177760          C.BUFF= -20             ;POINTER TO START OF BUFFER (0 IF NOT ALLOCATED)
1032          177762          C.BUFL= -16             ;BUFFER LENGTH FOR CHANNEL
1033          177764          C.CPTR= -14             ;POINTER TO NEXT BYTE IN BUFFER
1034          177766          C.CCNT= -12             ;NUMBER OF BYTES TO END OF BUFFER
1035          177770          C.WIDC= -10             ;NUMBER OF CHRS TO END OF LINE
1036          177772          C.WIDR= -6              ;RESET WIDTH VALUE
1037          177774          C.SRC=  -4              ;STATUS AREA FOR CHANNEL
1038          177776          C.RDLN= -2              ;LENGTH OF DATA READ
1039
1040                          ;BELOW ARE THE BITS DESCRIBING THE CHANNEL'S STATUS
1041
1043          020000                  C.FASW= 020000  ;File type zero
1044          010000                  C.OPN=  010000  ;THE CHANNEL IS OPEN
1045          004000                  C.FLSH= 004000  ;BUFFER NEEDS FLUSHING BEFORE INPUT
1046          002000                  C.REWT= 002000  ;BUFFER NEEDS TO BE REWRITTEN (RANDOM ACCESS)
1047          001000                  C.CZ=   001000  ;WRITE ^Z AND NULLS TO END OF BUFFER
1048          000400                  C.NOBF= 000400  ;SPECIAL BUFFERING, DON'T ALLOCATE OR DEALLOCATE
1049          000010                  C.RND=  FD.DIR  ;Well, sort of Random
1050          000004                  C.TT=   FD.TTY  ;Terminal Device
1051          000001                  C.FRC=  FD.REC  ;Character Oriented Device
```

```
                                .SBTTL  DEFINITIONS             RSX DATA

                                        .MCALL  FDBDF$, FDAT$A, FDRC$A, FDOP$A, NMBLK$, FSRSZ$
                                        .MCALL  FINIT$, FDAT$R, FDRC$R, FDBK$R, FDOP$R, FDBF$R
                                        .MCALL  OFID$,  OFNB$,  CLOSE$, WRITE$, READ$,  WTSE$
                                        .MCALL  OPEN$U, FDOF$L, EXIT$S, GPRT$S, GTIM$S, ASTX$S
                                        .MCALL  NBOF$L, DIR$,   QIOW$S, QIOW$,  QIOW$C, SFPA$S
                                        .MCALL  SVTK$S, ALUN$,  WAIT$,  GLUN$S, FDBK$A, OPNT$W
                                        .MCALL  GTSK$S, CSI$1,  CSI$2,  CSI$,   FCSBT$
                                        .MCALL  CSI$SW, CSI$ND, GET$S,  PUT$S

000000                                  FDOF$L          ;DEFINE THE FILE DESCRIPTOR BLOCK
000000                                  NBOF$L          ;DEFINE THE NAME BLOCK
000000                                  FCSBT$  DEF$L   ;DEFINE THE BITS
000000                                  FSRSZ$  0       ;Define the File Storage Region

        000036                  N.BLSZ= N.UNIT+2        ;SIZE OF FILE NAME BLOCK
        000140                  C.BLSZ= F.FNB+N.BLSZ    ;SIZE OF FDB/FNB
        000102                  C.ALOC= C.BLSZ-C.SBIT   ;TOTAL SIZE OF BLOCK

        000064                  C.BLOK= F.BKVB          ;NEXT BLOCK NUMBER TO READ/WRITE (2 WORDS)
        000024                  C.SRCP= F.BKST          ;POINTER TO I/O STATUS WORDS
        000102                  C.NAME= F.FNB           ;Name Block Pointer
        000010                  C.FLEN= F.EFBK          ;End-of-File Block Number
```

```
                        .SBTTL  DEFINITIONS     NAME BLOCKS

                        ;NAME BLOCKS CONTAIN THE FILE NAME IN DECODED FORMAT.  CURRENTLY
                        ;THE FORMAT IS VERY COMPATIBLE WITH THE FILES-11 FORMAT.
                        ;THE CURRENT FORMAT FOR A NAME IS:
                        ;
                        ;       DEVn:[prj,prg]NAME.EXT<prot>
                        ;
                        ;THE NAME BLOCK IS AS FOLLOWS:
                        ;
                        ;       ******************
                        ;       *                *              (N.FID)          <----- REFERENCE POINT
                        ;       * FILE ID FIELD *
                        ;       *                *
                        ;       ******************
                        ;       *                *      N.NAME  (N.FNAM)
                        ;       *   FILE NAME    *
                        ;       *                *
                        ;       ******************
                        ;       *   EXTENSION    *      N.EXT   (N.FTYP)
                        ;       ******************
                        ;       *  VERSION NO.   *              (N.FVER)
                        ;       ******************
                        ;       *  STATUS WORD   *      N.DESC  (N.STAT)
                        ;       ******************
                        ;       *                *              (N.NEXT)
                        ;       ******************
                        ;       *                *              (N.DID)
                        ;       *                *
                        ;       *                *
                        ;       ******************
                        ;       *   DEV NAME     *      N.DEV   (N.DVNM)
                        ;       ******************
                        ;       *   DEV NUMBER   *      N.DEVN  (N.UNIT)
                        ;       ******************
                        ;
                        ; Describe the RSX Filename Block Stuff
                        ;
000020                  N.DESC= N.STAT          ;Bits that Describe Name info picked up
000024                  N.PPN=  N.DID           ;UFD
000006                  N.NAME= N.FNAM          ;File Name
000014                  N.EXT=  N.FTYP          ;File Type
000032                  N.DEV=  N.DVNM          ;Device Name
000034                  N.DEVN= N.UNIT          ;Device Number
                        ;
                        ; Also the Name Block Bits
                        ;
000004                  ND.NAM= NB.NAM          ;Set if Explicit Name is Specified
000000                  ND.$NM= 0               ;Never have '?' Wildcard
000040                  ND.$N$= NB.SNM          ;Set if Wildcard Filename Specified
000002                  ND.EXT= NB.TYP          ;Set if Explicit Extension is Specified
000000                  ND.$EX= 0               ;Never have '?' Wildcard
000020                  ND.$E$= NB.STP          ;Set if Wildcard Extension is Specified
```

DEFINITIONS NAME BLOCKS

```
   1167          000100                  ND.PPN= NB.DIR          ;Set if UIC Specified
   1168          001400                  ND.$PN= NB.SD1!NB.SD2   ;Set if Wildcard in UIC
   1169          000200                  ND.DEV= NB.DEV          ;Set if Device Name Specified
```

```
                        .SBTTL  DEFINITIONS     MACROS

                        ;MACRO USED TO CALL SUBROUTINES

                        .MACRO CALL SUBR
                        .NLIST
                                JSR PC,SUBR
                        .LIST
                        .ENDM


                        ;MACRO USED TO CALL SUBROUTINES AND RETURN

                        .MACRO CALLR SUBR
                        .NLIST
                                JMP SUBR
                        .LIST
                        .ENDM


                        ;MACRO TO RETURN TO CALLER

                        .MACRO RETURN
                        .NLIST
                                RTS PC
                        .LIST
                        .ENDM


                        ;MACRO TO PUSH ITEMS ON SP STACK

                        .MACRO PUSH ARGS
                                .NLIST
                                .IRP ARG,<ARGS>
                                MOV ARG,-(SP)           ;PUSH ARG ON SP STACK
                                .ENDM
                                .LIST
                        .ENDM


                        ;MACRO TO POP ITEMS FROM SP STACK

                        .MACRO POP ARGS
                                .NLIST
                                .IRP ARG,<ARGS>
                                MOV (SP)+,ARG           ;POP ARG FROM SP STACK
                                .ENDM
                                .LIST
                        .ENDM

                        ;MACRO TO CONDITIONALLY INCREMENT BY 1

                        .MACRO CONINC ARG1,?L1
                        .NLIST
                                BNE L1                  ;DON'T INCREMENT IF 'Z' CLEAR
```

```
                        INC ARG1                ;INCREMENT
L1:
.LIST
.ENDM



;
; Macro to build a value frame on the SI stack
;
.MACRO SIFRAM ARG1,ARG2,ARG3
    .NLIST
    .IF B ARG3
        MOV     #SYSER,(SI)+            ;Return Address
    .IFF
        MOV     ARG3,(SI)+              ;Return address
    .ENDC

    .IF B ARG2
        MOV     #T.VAL+T.NUM,(SI)+      ;SI Frame Bits
    .IFF
        MOV     ARG2,(SI)+              ;SI Frame bits
    .ENDC

    .IF B ARG1
        MOV     R0,(SI)+                ;Value Pointer
    .IFF
        MOV     ARG1,(SI)+              ;Value pointer
    .ENDC
    .LIST
.ENDM SIFRAM



;MACROS TO FILL IN TABLE ENTRIES

.MACRO BFILL COUNT,ARG
.NLIST
.REPT COUNT
        .BYTE ARG
.ENDM
.LIST
.ENDM


.MACRO WFILL COUNT,ARG
.NLIST
.REPT COUNT
        .WORD ARG
.ENDM
.LIST
.ENDM


;MACRO TO BRING SI UP TO A WORD BOUNDARY

.MACRO SIEVEN
```

```
1726                                    .NLIST
1727                                        BIT #1,SI               ;SI ON WORD BOUNDARY?
1728                                        BEQ .+6                 ;YES, FINE
1729                                        MOVB #CHRFIL,(SI)+      ;NO, PUSH A FILL CHARACTER
1730                                    .LIST
1731                                .ENDM
1732
1733
1734                                ;MACRO TO GET NEXT INPUT CHARACTER
1735
1736                                .MACRO GETC ADDR
1737                                        DEC R3                                  ;ANY CHARACTERS LEFT?
1738                                        BLT ADDR                                ;NO, BRANCH TO ADDR
1739                                        MOVB (R4)+,R2                           ;YES, GET NEXT CHAR
1740                                .ENDM
1741
1742                                ;MACRO TO PRINT A MESSAGE TERMINATED BY A ZERO BYTE.
1743
1744                                .MACRO MPRINT ARG
1745                                    .NLIST
1746                                .IF NB,ARG
1747                                        PUSH R1                 ;SAVE R1
1748                                        MOV #ARG,R1             ;POINT TO ARG
1749                                .ENDC
1750                                        CALL PRINTM             ;PRINT THE MESSAGE
1751                                        POP R1                  ;RESTORE R1
1752                                    .LIST
1753                                .ENDM
1754
1755                                ;QUICK VERSION OF ABOVE (DOES NOT SAVE R1)
1756
1757                                .MACRO QPRINT ARG
1758                                .IF NB,ARG
1759                                        MOV #ARG,R1             ;POINT TO ARG
1760                                .ENDC
1761                                        CALL PRINTM             ;PRINT THE MESSAGE
1762                                .ENDM
1763
```

```
                        ;MACRO TO DEFINE A FLOATING POINT NUMBER OF PROPER LENGTH

                        .MACRO FLOAT ARGS
                        .IRP ARG,<ARGS>
                            .IF NE FIS
                                .FLT2   ARG
                            .IFF
                                .FLT4   ARG
                            .ENDC
                        .ENDR
                        .ENDM

                        ;MACROS TO SIMULATE INSTRUCTIONS NOT FOUND ON LOWER MACHINES

                        .MACRO SIMPSH SOURCE,REGNUM,SIGN,ROUTINE
                            .NLIST
                                .NTYPE REGTYP,<REGNUM>
                        .IF NE REGTYP&70
                        .ERROR NON-REGISTER USED IN REGISTER INSTRUCTION
                        .ENDC
                                PUSH <SOURCE,#SIGN*REGTYP>
                                CALL ROUTINE
                            .LIST
                        .ENDM



                        .MACRO SXT DST,?LOC1,?LOC2
                            .NLIST
                                BMI LOC1
                                CLR DST
                                BR LOC2
                        LOC1:   MOV #-1,DST
                        LOC2:
                            .LIST
                        .ENDM


                        .MACRO SOB REG,LOC1
                            .NLIST
                                DEC REG
                                BNE LOC1
                            .LIST
                        .ENDM

                        .MACRO MUL SRC1,REG
                                SIMPSH SRC1,REG,1,IMUL
                        .ENDM

                        .MACRO DIV SRC1,REG
                                SIMPSH SRC1,REG,1,IDIV
                        .ENDM
```

```
1823                                    .MACRO ASH SRC1,REG
1824                                            SIMPSH SRC1,REG,-1,IASH
1825                                    .ENDM
1826
1827
1828                                    .MACRO ASHC SRC1,REG
1829                                            SIMPSH SRC1,REG,1,IASHC
1830                                    .ENDM
1831
1832
1833                                    .MACRO XOR SRC1,REG
1834                                            SIMPSH SRC1,REG,-1,IXOR
1835                                    .ENDM
1836
1838
1840                                    ;FLOATING POINT FAKE-OUT MACROS
1841
1842                                    ;TYPE 1:  AC AND FARG
1843                                    .IRP INS,<ADD,LD,SUB,CMP,MUL,MOD,ST,DIV,LDCF,STCF,LDCD,STCD>
1844                                      .MACRO INS'F FSRC,FDST
1845                                        .NLIST
1846                                            JSR R0,$'INS'F
1847                                            MOV FSRC,FDST
1848                                        .LIST
1849                                      .ENDM
1850                                      .MACRO INS'D FSRC,FDST
1851                                        .NLIST
1852                                            JSR R0,$'INS'D
1853                                            MOV FSRC,FDST
1854                                        .LIST
1855                                      .ENDM
1856                                    .ENDR
1857
1858                                    ;TYPE 2:  FARG
1859                                    .IRP INS,<CLR,TST,ABS,NEG>
1860                                      .MACRO INS'F FARG
1861                                            .NLIST
1862                                            JSR R0,$'INS'F
1863                                            MOV R0,FARG
1864                                            .LIST
1865                                      .ENDM
1866                                      .MACRO INS'D FARG
1867                                            .NLIST
1868                                            JSR R0,$'INS'D
1869                                            MOV R0,FARG
1870                                            .LIST
1871                                      .ENDM
1872                                    .ENDR
1873
1874                                    ;TYPE 3:  AC AND DIFFERENT FORMAT ARG OR FARG
1875                                    .IRP INS,<LDEXP,LDCIF,LDCLF,LDCID,LDCLD,STEXP,STCFI,STCFL,STCDI,STCDL>
1876                                      .MACRO INS SRC1,DST
1877                                            .NLIST
1878                                            JSR R0,$'INS
1879                                            MOV SRC1 DST
1880                                             LIST
1881                                      .EN
```

```
1882                                    .ENDR
1883
1884                                    ;TYPE 4:  IARG
1885                                    .IRP INS,<LDFPS,STFPS,STST>
1886                                      .MACRO INS ARG
1887                                            .NLIST
1888                                            JSR R0,$'INS
1889                                            MOV R0,ARG
1890                                            .LIST
1891                                      .ENDM
1892                                    .ENDR
1893
1894                                    ;TYPE 5:  NILADIC
1895                                    .IRP INS,<CFCC,SETF,SETI,SETD,SETL>
1896                                      .MACRO INS
1897                                            .NLIST
1898                                            JSR R0,$'INS
1899                                            .LIST
1900                                      .ENDM
1901                                    .ENDR
1902
```

```
                                .SBTTL  DEFINITIONS     APL CHARACTERS

                                ; THIS TABLE CONTAINS THE ASCII DEFINITIONS FOR CERTAIN APL
                                ; CHARACTERS.

000003                          CHR.CC  =003    ;CONTROL-C
000007                          CHRBEL  =007    ;BELL
000010                          CHR.BS  =010    ;BACKSPACE
000011                          CHR.TB  =011    ;TAB
000012                          CHR.LF  =012    ;LINE FEED
000015                          CHR.CR  =015    ;CARRIAGE RETURN
000016                          CHR.SO  =016    ;SHIFT OUT
000017                          CHR.SI  =017    ;SHIFT  IN
000032                          CHR.CZ  =032    ;CONTROL Z
000040                          CHRSPA  =040    ;SPACE
000042                          CHRRPR  =042    ;RIGHT PARENTHESIS
000042                          ASCQUO  =042    ;ASCII QUOTATION MARK
000043                          CHR.LT  =043    ;<
000046                          CHR.GT  =046    ;>
000047                          CHRRBR  =047    ;RIGHT BRACKET
000051                          CHRAND  =051    ;AND
000053                          CHRDIV  =053    ;DIVIDE
000054                          CHRCMA  =054    ;COMMA
000055                          CHRPLU  =055    ;PLUS SIGN
000056                          CHRDOT  =056    ;DECIMAL POINT
000057                          CHR.SL  =057    ;SLASH
000060                          CHR..0  =060    ;ZERO
000061                          CHR..1  =061    ;ONE
000072                          CHRLPR  =072    ;LEFT PARENTHESIS
000073                          CHRLBR  =073    ;LEFT BRACKET
000074                          CHRSEM  =074    ;SEMI COLON
000075                          CHRMUL  =075    ;MULTIPLY
000076                          CHRCOL  =076    ;COLON
000100                          CHRNEG  =100    ;NEGATE SIGN
000105                          CHREP   =105    ;EPSILON
000106                          CHR.UL  =106    ;UNDERLINE
000107                          CHRDEL  =107    ;DEL
000110                          CHRDLT  =110    ;DELTA
000112                          CHRSMO  =112    ;SMALL 'O'
000113                          CHRQUO  =113    ;QUOTE MARK
000114                          CHRBOX  =114    ;QUAD
000120                          CHRPWR  =120    ;POWER (STAR)
000121                          CHRQST  =121    ;QUESTION MARK
000131                          CHRUPA  =131    ;UP ARROW
000133                          CHRBKR  =133    ;BACK ARROW
000135                          CHRFRO  =135    ;FOREWARD ARROW (BRANCH)
000137                          CHRMNS  =137    ;MINUS SIGN (SUBTRACTION OPERATOR)
000141                          CHR..A  =141    ;A
000145                          CHR..E  =145    ;E
000163                          CHR..S  =163    ;S
000164                          CHR..T  =164    ;T
000165                          CHR..U  =165    ;U
000172                          CHR..Z  =172    ;Z
000176                          CHRDLR  =176    ;DOLLAR SIGN
000177                          CHRFIL  =177    ;FILL CHARACTER
000200                          CHRBOL  =200    ;BEGINNING OF CODESTRING
000271                          CHRPDL  =271    ;PROTECT DEL
```

DEFINITIONS APL CHARACTERS

```
1962
1963          177400                  UPPER   =^B1111111100000000     ;MASK TO KILL UPPER BYTE
1964
1965
```

```
                                .SBTTL  DEFINITIONS     ERRORS

                                ;ALL ERRORS ARE DEFINED HERE AS TRAPS.  TO CALL AN ERROR, SIMPLY USE
                                ;THE ERROR'S NAME SUCH AS:
                                ;
                                ;       SYSERR
                                ;
                                ;WHICH IS USED TO SIGNIFY A SYSTEM ERROR.  ANY ADDITIONS OR DELETIONS
                                ;HERE SHOULD BE NOTED IN THE FILE 'APL2' IN THE ERROR TEXT
                                ;DEFINITIONS.


                                .MACRO ERR NAME,NUMBER
                                        .IF NB NUMBER
                                          TMPTAG=NUMBER
                                        .ENDC
                                        NAME=104400+TMPTAG
                                             NAME=   NAME
                                        TMPTAG=TMPTAG+1
                                .ENDM

                000000          TMPTAG=0                        ;ERRORS START AT 0

000000                          ERR     SYSERR, 2.              ;SYSTEM ERROR
000000                          ERR     WSFULL, 3.              ;WORK SPACE FULL
000000                          ERR     SYMTBF, 4.              ;SYMBOL TABLE FULL
000000                          ERR     DEFERR, 5.              ;DEFN ERROR
000000                          ERR     LABERR, 6.              ;LABEL ERROR
000000                          ERR     SYNERR, 7.              ;SYNTAX ERROR
000000                          ERR     INDERR, 8.              ;INDEX ERROR
000000                          ERR     RANERR, 9.              ;RANK ERROR
000000                          ERR     LENERR, 10.             ;LENGTH ERROR
000000                          ERR     VALERR, 11.             ;VALUE ERROR
000000                          ERR     DPTERR, 14.             ;DEPTH ERROR
000000                          ERR     DOMERR, 15.             ;DOMAIN ERROR
000000                          ERR     STOP,   16.             ;EXECUTION STOP
000000                          ERR     COMERR, 21.             ;COMMAND ERROR
000000                          ERR     CHRERR, 25.             ;CHARACTER ERROR
000000                          ERR     NYIERR, 26.             ;NOT YET IMPLEMENTED
000000                          ERR     SIDAMN, 27.             ;SI DAMAGE
000000                          ERR     TURKEY, 30.             ;TURKEY ERROR (GOBBLE!! GOBBLE!!)
000000                          ERR     BADERR, 31.             ;BAD ERROR - SHOULD NEVER HAPPEN
000000                          ERR     RESERR, 32.             ;RESTART ERROR

                                ; BEGINNING OF I/O ERRORS

                000062          I.OERR=50.                      ;START OF I/O ERRORS
000000                          ERR     INSNAM, I.OERR          ;Installation Name
000000                          ERR     BADDIR, I.OERR          ;Bad Directory
000000                          ERR     ILFNAM, I.OERR+2.       ;Illegal File Name
000000                          ERR     DIRFUL, I.OERR+4.       ;Directory Full
000000                          ERR     FILNTF, I.OERR+5.       ;File Not Found
000000                          ERR     DEVNTF, I.OERR+6.       ;Device Not Found
000000                          ERR     CHNOPN, I.OERR+7.       ;File Already Open
000000                          ERR     NOTAVL, I.OERR+8.       ;Device Not Available
000000                          ERR     NOTOPN, I.OERR+9.       ;Channel not Open
```

```
2024 000000                    ERR     EOF,    I.OERR+11.      ;End of File on Device
2025 000000                    ERR     IOERR,  I.OERR+13.      ;I/O Error
2026 000000                    ERR     ILLCHN, I.OERR+46.      ;Illegal Channel Number
2027 000000                    ERR     LNLNG,  I.OERR+47.      ;Line too Long
2028 000000                    ERR     ILLSW,  I.OERR+67.      ;Illegal switch usage
2029
2030          000166           ERRMAX= TMPTAG                  ;MAXIMUM ERROR POSSIBLE
```

```
                                        ;++
                                        ; DEFINE DATA AREA
                                        ;
                                        ; ALL DATA IS DEFINED AS A LOCATION IN THE CSECT, 'APLVAR' AND THE
                                        ; VALUE GIVEN WITH IT IS PUT IN THAT LOCATION.  IF NO VALUE IS
                                        ; GIVEN, ALL BYTES ARE ZEROED OUT.
                                        ;       IF THE LEN IS           THE VALUE IS TAKEN AS
                                        ;           1 BYTE                      BYTE
                                        ;           2 BYTES                     WORD
                                        ;           FPPLEN                      FLOATING POINT #
                                        ;--

000000                                          .PSECT  DEF$0,  RW, D, OVR, GBL
000000                                          .PSECT  DEF$1,  RW, D, OVR, GBL
000000                                          .PSECT  DEF$2,  RW, D, OVR, GBL

                                        .MACRO  $$DEF   LEN,    VALUE
                                            .IF B <VALUE>
                                                .IF NB <LEN>
                                                    .NLIST
                                                    .REPT LEN
                                                        .BYTE 0
                                                    .ENDR
                                                    .LIST
                                                .ENDC
                                            .IFF
                                                .IF EQ LEN-1
                                                    .BYTE VALUE
                                                .IFF
                                                    .IF EQ LEN-2
                                                        .WORD VALUE
                                                    .IFF
                                                        .IF EQ LEN-8.
                                                            .FLT4 VALUE
                                                        .IFF
                                                            .ERROR      ;VALUE IS ILLEGAL FOR SIZE OF NAME
                                                        .ENDC
                                                    .ENDC
                                                .ENDC
                                            .ENDC
                                        .ENDM   $$DEF


                                                .MACRO  $DEF    NAME,   LEN,    VALUE
                                                    .IF NB NAME
                                        NAME:
                                                        .IF NE .DEF.-2
                                                            .GLOBL      NAME
                                                        .ENDC
                                                    .ENDC
                                                    $$DEF       LEN,    <VALUE>
                                                .ENDM   $DEF
```

```
                000000                          .DEF.=  0               ;Special Case
                                        ;+
                                        ; Channels 13 and 14 (the Utility Channels) are very special and
                                        ; must not be modified while SAVE or LOAD is in progress. Hence,
                                        ; they are in a special place that is not LOADed.
                                        ;-
000000                                          .PSECT  DEF$0

000000                                  $DEF    ,       -C.SBIT         ;STUFF FOR UTIL CHANNEL
000022                                  $DEF    CHAN13. <C.BLSZ-C.SBIT> ;Stuff for Utility Channel 1.
000204                                  $DEF    CHAN14. C.BLSZ          ;Stuff for Utility Channel 2.
000344                                  $DEF    UTNAME. N.BLSZ          ;Name Block for COPY

000402                                  $DEF    OMODE.  1.      377     ;OLD 'TMODE' FOR FILES
000403                                  $DEF    TMODE.  1.      1       ;0=APL TERMINAL, ELSE TTY
000404                                  $DEF    .MOLUN. 2               ;# OF MESSAGE OUTPUT

                                        ;+
                                        ; Some of these first variables are very special and should not be
                                        ; cleared with a ')CLEAR'
                                        ;-
                000001                          .DEF.=  .DEF.+1         ;NON-OVERLAID STUFF
000000                                          .PSECT  DEF$1

000000                                  $DEF    SYSTAT. 2.      0       ;SYSTEM STATUS
000002                                  $DEF    SPINIT. 2.      0       ;SP RESET VALUE
000004                                  $DEF    SPHOLD. 2.      0       ;STACK POINTER SLOT DURING LOAD/SAVE

                000002                          .DEF.=  .DEF.+1         ;OVERLAID STUFF
000000                                          .PSECT  DEF$2

                                        ;+
                                        ; These other variables need to be reset by the ')CLEAR' command
                                        ;-
000000                                  $DEF    VARBLS. 0                       ;START OF VARIABLES
000000                                  $DEF    VERNUM. 2.      VRSION*256.+EDIT;Version and edition
000002                                  $DEF    LOWCOR. 2.      0               ;Pointer to beginning of user's WS
000004                                  $DEF    ENDCOR. 2.      0               ;Pointer to the top of the WS
000006                                  $DEF    DIGITS. 2.      6+<FPPLEN/10*4.>;INITIAL # OF DIGITS
000010                                  $DEF    ROUND.  8..     5E-6            ;ROUNDING CONSTANT (SINGLE PRECISION)
000020                                  $DEF    WROUND. 8..     5E-7            ;ROUNDING CONSTANT FOR WRITING FILES
000030                                  $DEF    ORIGIN. 2.      1               ;SUBSCRIPTING ORIGIN
000032                                  $DEF    FORIGN. 8..     1.0             ;FLOATING ORIGIN
000042                                  $DEF    FPMODE. 2.      <<FPPLEN&10>*20+1000>;INITIAL FPP MODE
000044                                  $DEF    SIZTAB. 2.      0               ;TABLE FOR SHIFTS
000046                                  $DEF    .       2.      0               ;BOOLEAN ENTRY
000050                                  $DEF    .       2.      1               ;INTEGER ENTRY
000052                                  $DEF    PRECSN. 2.      FPPLEN/10+2     ;SHIFT TO MULTIPLY BY PRECISION
000054                                  $DEF    LENTAB. 2.      1               ;TABLE FOR SIZE LOOKUP
000056                                  $DEF    .       2.      1               ;BOOLEAN ENTRY
000060                                  $DEF    .       2.      2               ;WORD ENTRY
000062                                  $DEF    NUMLEN. 2.      FPPLEN          ;LENGTH OF FLOATING NUMBERS
000064                                  $DEF    INPTR.  2.      GETL            ;POINTER TO INPUT ROUTINE
000066                                  $DEF    OUTPTR. 2.      OCHAR           ;POINTER TO OUTPUT ROUTINE
000070                                  $DEF    LWIDTH. 2.      72.             ;PAPER WIDTH
000072                                  $DEF    WIDTH.  2.      72.             ;PAPER WIDTH
000074                                  $DEF    WSNAME. 0                       ;WS NAME
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 39-1
DEFINITIONS     ERRORS

   2206 000074                          $DEF        ,       N.BLSZ                  ;LENGTH IS SIZE OF NAME BLOCK
   2207 000132                          $DEF    WSTIME,     2.      0               ;WSTIME (RT-11 LIKE FORMAT)
   2208 000134                          $DEF        ,       2.      0               ;
   2209 000136                          $DEF    WSDATE,     2.      0               ;WSDATE    ''    ''    ''
   2210 000140                          $DEF        ,       2.      0               ;
   2212 000142                          $DEF    FUZZ,       8.,     5E-7            ;FUZZ FACTOR
   2213 000152                          $DEF    CTEPS,      8.,     5E-7            ;EPS Constant, Plus Argument, =Fuzz
   2214 000162                          $DEF        ,       8.,     0.500000250000125E-7;Minus Argument
   2215 000172                          $DEF    CTRMAX,     8.,     0.5000001250003125;RMAX Constant, Plus argument
   2216 000202                          $DEF        ,       8.,     0.4999968749987499999;Minus Argument
   2224 000212                          $DEF    CTMAX,      8.,     0.381966011250105;Maximum Value for Fuzz
   2225
   2227 000222                          $DEF    FLSWTB,     10.                     ;Formatted ASCII switch table
   2229 000234                          $DEF        ,       -C.SBIT                 ;SPACE FOR USER TERMINAL
   2230 000256                          $DEF    CHAN00,     0                       ;USER TERMINAL
   2232 000256                          $DEF        ,       24.                     ;THIS IS THE AREA
   2236 000306                          $DEF        ,       -C.SBIT                 ;1ST USER CHANNEL
   2237 000330                          $DEF    CHAN01,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 1
   2238 000512                          $DEF    CHAN02,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 2
   2239 000674                          $DEF    CHAN03,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 3
   2240 001056                          $DEF    CHAN04,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 4
   2241 001240                          $DEF    CHAN05,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 5
   2242 001422                          $DEF    CHAN06,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 6
   2243 001604                          $DEF    CHAN07,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 7
   2244 001766                          $DEF    CHAN08,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 8
   2245 002150                          $DEF    CHAN09,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 9
   2246 002332                          $DEF    CHAN10,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 10
   2247 002514                          $DEF    CHAN11,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 11
   2248 002676                          $DEF    CHAN12,     <C.BLSZ-C.SBIT>         ;USER CHANNEL 12
   2249
   2253
   2254 003060                          $DEF    CHANEX,     0                       ;Execute Channel
   2255
   2257 003060                          $DEF    REENTR,     2.      1               ;FAKE FPP RE-ENTERED FLAG
```

DEFINITIONS ERRORS

```
2260 003062                    $DEF    RANDY.  2.      1       ;RANDOM KERNEL
2261 003064                    $DEF    RANDAC. 8..             ;BUFFER USED TO GENERATE RANDOM NUMBERS
2262 003074                    $DEF    FIXPTR. 2               ;POINTER TO INPUT BUFFER
2263 003076                    $DEF    FXPTR1. 2               ;POINTER TO RIGHTMOST CHR OF LINE
2264 003100                    $DEF    LEXPTR. 2               ;POINTER TO START OF LEX LINE IN SI
2265 003102                    $DEF    GNOME.  2               ;INPUT CHR POINTER
2266 003104                    $DEF    SUBCNT. 2               ;COUNT OF DIMENSIONS IN ARRAY
2267 003106                    $DEF    SUBDCT. 2               ;COUNT OF DIMS IN RESULT
2268 003110                    $DEF    SUBPTR. 2               ;POINTER TO ARRAY'S DIM LIST
2269 003112                    $DEF    DIMBLK. 2               ;POINTER TO DIMENSION ARG BLOCK
2270 003114                    $DEF    DIMPTR. 2               ;POINTER TO FIRST DIM ARG
2271 003116                    $DEF    DIMTYP. 2               ;TYPE OF DIM ARG
2272 003120                    $DEF    MONBLK. 2               ;POINTER TO MONADIC ARG BLOCK
2273 003122                    $DEF    MONPTR. 2               ;POINTER TO MONADIC ARG
2274 003124                    $DEF    MONTYP. 2               ;TYPE OF RESULT (CHAR/BOOL/INT/FLOAT)
2275 003126                    $DEF    MONSFT. 2               ;MONADIC SHIFT CONSTANT
2276 003130                    $DEF    MONLEN. 2               ;MONADIC LENGTH CONSTANT
2277 003132                    $DEF    MONCVT. 2               ;POINTER TO CONVERSION ROUTINE
2278 003134                    $DEF    MONVAL. 2               ;VALUE TEMPORARY
2279 003136                    $DEF    RESULT. 2               ;POINTER TO RESULT BLOCK
2280 003140                    $DEF    RESPTR. 2               ;RESULT DATA POINTER
2281 003142                    $DEF    RESBIT. 2               ;RESULT DESCRIPTION BITS
2282 003144                    $DEF    RESTYP. 2               ;TYPE OF RESULT
2283 003146                    $DEF    RESSFT. 2               ;RESULT SHIFT CONSTANT
2284 003150                    $DEF    RESLEN. 2               ;RESULT LENGTH CONSTANT
2285 003152                    $DEF    RESCVT. 2               ;CONVERSION ROUTINE FOR RESULT
2286 003154                    $DEF    RESVAL. 2               ;VALUE TEMPORARY
2287 003156                    $DEF    COMRTN. 2               ;POINTER TO COMPARE ROUTINE
2288 003160                    $DEF    LARGFL. 2               ;SET IF LEFT ARG NOT SCALAR
2289 003162                    $DEF    LARGBL. 2               ;POINTER TO LEFT ARG BLOCK
2290 003164                    $DEF    LARGPT. 2               ;POINTER TO LEFT ARG
2291 003166                    $DEF    LARGLN. 2               ;VOLUME TO LEFT ARG
2292 003170                    $DEF    LARGTY. 2               ;RIGHT ARG TYPE
2293 003172                    $DEF    LARGSF. 2               ;LEFT ARG SHIFT CONSTANT
2294 003174                    $DEF    LARGLE. 2               ;LEFT ARG LENGTH CONSTANT
2295 003176                    $DEF    LARGCV. 2               ;RIGHT ARG CONVERSION ROUTINE
2296 003200                    $DEF    LARGVA. 2               ;TEMPORARY FOR VALUE
2297 003202                    $DEF    RARGFL. 2               ;SET IF RIGHT ARG 1. COMP ARRAY
2298 003204                    $DEF    RARGBL. 2               ;POINTER TO RIGHT ARG BLOCK
2299 003206                    $DEF    RARGPT. 2               ;POINTER TO RIGHT ARG
2300 003210                    $DEF    RARGLN. 2               ;VOLUME OF RIGHT ARG
2301 003212                    $DEF    RARGTY. 2               ;LEFT ARG TYPE
2302 003214                    $DEF    RARGSF. 2               ;RIGHT ARG SHIFT CONSTANT
2303 003216                    $DEF    RARGLE. 2               ;RIGHT ARG LENGTH CONSTANT
2304 003220                    $DEF    RARGCV. 2               ;LEFT ARG CONVERSION ROUTINE
2305 003222                    $DEF    RARGVA. 2               ;TEMPORARY FOR VALUE
2306 003224                    $DEF    DYDCNT. 2               ;COUNT OF ELEMENTS IN ARRAY
2307 003226                    $DEF    NUMWID. 2               ;NUMERIC FIELD WIDTH
2308 003230                    $DEF    NUMFLG. 0               ;SAVE NUMDFL AND NIMRIG IN THE SAMME WORD
2309 003230                    $DEF    NUMDFL. 1               ;SET IF DECIMAL POINT FOUND
2310 003231                    $DEF    NUMRIG. 1               ;COUNT OF DIGITS ON RIGHT OF DECIML POINT
2311 003232                    $DEF    NUMNEG. 1               ;SET IF NUMBER NEGATIVE
2312 003233                    $DEF    NUMEFL. 1               ;SET IF EXPONENT FOUND
2313 003234                    $DEF    NUMDFL. 1               ;SET SET IF DIGIT FOUND
2314 003235                    $DEF    NUMZFL. 1               ;SET IF NON-ZERO DIGIT FOUND
2315 003236                    $DEF    NUMESN. 1               ;SET IF EXPONENT NEGATIVE
2316 003237                    $DEF    .       1               ;<FILLER FOR EVENING COUNTER>
```

```
2317 003240                    $DEF    NAMBUF, 0               ;SHARED NAME BUFFER FOR DECODING NAME
2318 003240                    $DEF    ,       10              ;FAKE ZEROES
2319           000010                  CMDMAX= 10              ;MAX # OF CHARS ALLOWED IN A COMMAND
2320 003250                    $DEF    CMDBUF, 0               ;COMMAND BUFFER (SHARED WITH NUM OUTPUT)
2321 003250                    $DEF    INNTMP, 0               ;INNER PRODUCT TEMPORARY
2322 003250                    $DEF    FLCBUF, 20              ;FLOATING OUTPUT BUFFER
2323 003270                    $DEF    FLOEXP, 2               ;FLOATING EXPONENT
2324 003272                    $DEF    FLONEG, 2               ;SIGN OF NUMBER
2325 003274                    $DEF    ALLINT, 2               ;SET FOR CLOSE PACKING OF NUMBERS
2326 003276                    $DEF    ARYRET, 2               ;POINTER TO RETURN FROM ROUTINE
2327 003300                    $DEF    NINPT2, 2               ;LORD KNOWS WHAT THIS POINTER DOES
2328 003302                    $DEF    NINPT1, 2               ;THIS ONE TOO...
2329 003304                    $DEF    NINSTK, 2               ;PTR PAST INDEX FRAMES
2330 003306                    $DEF    NINTYP, 2               ;NON-ZERO MEANS NUMERIC ARRAY
2331 003310                    $DEF    NINTMP, 2               ;TEMP STORAGE
2332 003312                    $DEF    ROVER,  2               ;POINTER TO FREE STORAGE
2333 003314                    $DEF    SIFRAM, 2               ;POINTER TO CURRENT SI STACK FRAME
2334           000621                  LBUFLN=401.             ;SIZE OF LINE BUFFER (ODD)
2335 003316                    $DEF    LBUFF,  LBUFLN+1        ;LINE BUFFER
2336           000622                  TTYLEN=402.             ;SIZE OF I/O BUFFER
2337 004140                    $DEF    TTYBUF, TTYLEN          ;I/O BUFFER
2338 004762                    $DEF    NPRT1,  2               ;POINTER TO FIRST ITEM TO PRINT
2339 004764                    $DEF    FLAGS,  2               ;CURRENT FLAGS FOR THIS FRAME
2340 004766                    $DEF    LSTERP, 2               ;VALUE OF WIDTH AT ERROR
2341 004770                    $DEF    ERRPTR, 2               ;POINTER TO ERROR CALLER
2342 004772                    $DEF    ERRNUM, 2               ;ERROR NUMBER OF LAST ERROR
2343 004774                    $DEF    ERRO,   16              ;ERROR REGISTER BUFFER
2344 005012                    $DEF    FHEADR, 2               ;POINTER TO FUNCTION HEADER
2345 005014                    $DEF    FLINE,  2               ;POINTER TO CURRENT LINE BLOCK
2346 005016                    $DEF    INNTM2, 0               ;Inner product temporary result
2347 005016                    $DEF    LINE,   8.              ;CURRENT LINE NUMBER
2348 005026                    $DEF    OPTM14, 0               ;Inner product temporary
2349 005026                    $DEF    LININC, 8.              ;CURRENT LINE INCREMENT
2350 005036                    $DEF    LOCLCT, 2               ;COUNT OF LOCAL VARIABLES
2351 005040                    $DEF    FHRET,  2               ;POINTER TO SUBROUTINE RETURN
2352 005042                    $DEF    LABLPT, 2               ;POINTER TO LABELS SYMBOL BLOCK
2353 005044                    $DEF    FNINSI, 2               ;NON-ZERO IF FN IS IN SI STACK
2354 005046                    $DEF    BRSELF, 2               ;SPECIAL FLAG TO ALLOW A BRANCH TO
2358                                                           ;EXECUTE A LINE WITH A STOP BIT SET.
2362                           ;I/O VARIABLES
2363
2364           001000                  OBUFLN=512.
2365 005050                    $DEF    ICHAN,  2               ;POINTER TO INPUT CHANNEL BLOCK
2366 005052                    $DEF    OCHAN,  2               ;POINTER TO OUTPUT CHANNEL BLOCK
2367 005054                    $DEF    EXEBUF, 2               ;POINTER TO EXECUTE OP BUFFER
2368 005056                    $DEF    ESCT,   2               ;EXECUTE CHARACTER COUNT
2369 005060                    $DEF    ESPT,   2               ;EXECUTE CHARACTER POINTER
2370 005062                    $DEF    EDIFLG, 1               ;1=LINE EDIT MODE, -1=CHR SEARCH
2371 005063                    $DEF    NEGSGN                  ;MINUS SIGN CHARACTER FOR NUMOUT
2372 005063                    $DEF    EDICHR, 1               ;EDIT SEARCH CHARACTER
2373 005064                    $DEF    EDICNT, 2               ;EDIT CHARACTER SKIP COUNT
2374
2375                                   ;END OF NON-INITIALIZED I/O VARIABLES
```

```
2377 005066                     $DEF    LFTOP
2378 005066                     $DEF    REDIGN
2379 005066                     $DEF    TKFRAM
2380 005066                     $DEF    RLTBAS
2381 005066                     $DEF    OPTMP0, 2               ;TEMPORARY OP VARS FOR EACH OP TO USE
2382 005070                     $DEF    RTOP
2383 005070                     $DEF    REDCNT
2384 005070                     $DEF    TKLEFT
2385 005070                     $DEF    ROTINC
2386 005070                     $DEF    OPTMP1, 2
2387 005072                     $DEF    LITOP
2388 005072                     $DEF    REDOFF
2389 005072                     $DEF    TKDIMS
2390 005072                     $DEF    CCNT
2391 005072                     $DEF    OPTMP2, 2
2392 005074                     $DEF    RBASE
2393 005074                     $DEF    OPTMP3, 2
2394 005076                     $DEF    COMBINE
2395 005076                     $DEF    OPTMP4, 2
2396 005100                     $DEF    LBINC
2397 005100                     $DEF    OPTMP5, 2
2398 005102                     $DEF    LINC
2399 005102                     $DEF    OPTMP6, 2
2400 005104                     $DEF    RINC
2401 005104                     $DEF    OPTMP7, 2
2402 005106                     $DEF    LCOUNT
2403 005106                     $DEF    OPTMP8, 2
2404 005110                     $DEF    RCOUNT
2405 005110                     $DEF    OPTMP9, 2
2406 005112                     $DEF    RCOUNR
2407 005112                     $DEF    OPTM10, 2
2408
2409
2410 005114                     $DEF    DEVIL, 2                ;INCREMENT FOR RESULT POINTER
2411 005116                     $DEF    BLZBUB, 2               ;PTR TO DYADIC OP ROUTINE (OPTIONAL)
2412 005120                     $DEF    CORPTR, 2               ;CURRENT CORE PEEK/POKE ADDRESS
2413 005122                     $DEF    COMPTR, 2               ;POINTER TO DIM BEING USED
2414 005124                     $DEF    COMSKP, 2               ;NUMBER OF ELEMENTS TO SKIP AT A TIME
2415 005126                     $DEF    COMCNT, 2               ;NUMBER OF GROUPS TO DO AT A TIME
2416 005130                     $DEF    CNTSIZ, 2               ;SIZE OF 'COMCNT' GROUPS
2417 005132                     $DEF    ELESIZ, 2               ;SIZE OF EACH ELEMENT
2418 005134                     $DEF    SKPSIZ, 2               ;SIZE OF ELEMENTS FOR EACH LEVEL
```

```
                                                ;SYMBOL TABLE AND HASH TABLE DEFINITIONS
                                                
000200                                  HSHSIZ= 128.                    ;HASH TABLE SIZE
177601                                  HSHMSK= ^C<HSHSIZ-2>            ;MASK FOR HASHING FUNCTION
005136                          $DEF    HASHC,  2                       ;HASH FUNCTION VALUE WORD
005140                          $DEF    HASHTB, HSHSIZ                  ;HASH TABLE
                                
005340                          $DEF    SYMF.   2.      SYMTAB          ;POINTER TO SYMBOL TABLE TOP
005342                          $DEF    ,       2.      0               ;ALPHABETIC LINK FOR SYMB1
005344                          $DEF    SYMB1,  2.      0               ;ALPHABETIC LIST HEADER
005346                          $DEF    SYMPT1, 2                       ;POINTER TO SYMBOL TO ENTER INTO TABLE
                                
000774                                  STKINC= 508.                    ;SIZE OF TABLE OR STACK INCREMENT
000054                                  SYMWRN= 44.                     ;SIZE OF SYMBOL TABLE WARNING AREA
005350                          $DEF    SYMTOP, 2,      SYMTP           ;POINTER TO SYMBOL TABLE WARNING AREA
005352                          $DEF    SYMTAB, STKINC-SYMWRN           ;SYMBOL TABLE
006272                          $DEF    SYMTP,  SYMWRN                  ;SYMBOL TABLE WARNING AREA
                                
                                                ;SI STACK DEFINITION
                                
000200                                  SIWARN= 128.                    ;SIZE OF SI STACK WARNING AREA
006346                          $DEF    SITOP,  2,      SITP            ;POINTER TO SI STACK WARNING AREA
006350                          $DEF    SISTAK, STKINC-SIWARN           ;SI STACK
007144                          $DEF    SITP,   SIWARN                  ;WARNING LIMIT
```

```
2446 007344                             $DEF    PATCH.  20              ;PATCH AREA
2447
2448 007364                             $DEF    NULL.   2.      0       ;NULL ARRAY IMAGE
2449 007366                             $DEF    .       1.      1       ;# OF DIMENSIONS
2450 007367                             $DEF    .       1.      A.ASB+A.NUM+A.CON+A.TMP
2451 007370                             $DEF    .       2.      0       ;ALEN
2452 007372                             $DEF    .       2.      0       ;DIM 1
2453
2454                                    ;
2455                                    ; HERE IS BUILT A FREE BLOCK REPRESENTING ALL OF CORE WHEN APL IS
2456                                    ; INITIALIZED
2457                                    ;
2458 007374                             $DEF    .       4               ;BLOCK HEADER
2459 007400                             $DEF    FREBLK. 0               ;REST OF MINIMUM BLOCK
2460 007400                             $DEF    VARBLE. 0               ; END OF VARIABLE AREA
2461
2464                                            .MACRO  INITCH  CHANUM
2465                                                .IRP CHANNO,<CHANUM>
2466                                                    .=      CHAN'CHANNO
2467                                                    FDBDF$
2468                                                    FDBK$A  ,,,32.,CHAN'CHANNO+C.SRC
2469                                                    FDOP$A  CHANNO'.
2470                                                .ENDR
2471                                            .ENDM
2472
2473 000406                                     .PSECT  DEF$0
2474 000406                                     INITCH  <13,14>
2475            000022'                         .=CHAN13
2476 000022                                             FDRC$A  FD.RWM
2477 000022                                             FDAT$A  R.FIX
2478            000204'                         .=CHAN14
2479 000204                                             FDRC$A  FD.RWM
2480 000204                                             FDAT$A  R.FIX
2481
2482 007400                                     .PSECT  DEF$2
2483 007400                                     INITCH  <01,02,03,04,05,06,07,08,09,10,11,12>
2484
2485            000222'                         .=FLSWTB
2486 000222                                             CSI$SW  FA,C.FASW,,,NEG,,LONG
2487 000232                                             CSI$ND
2512
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 45
DEFINITIONS     ERRORS

      1         000001                          .END
```

APLDF MACRO M1113 08-FEB-80 08:58 PAGE 45-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.ALOC= 000162    D.CTLM= 000000    EDIFLG  005062R   025
AC1    =%000001          CHRLPR= 000072    C.BLOK= 000064    D.DEAL= 000003    EDIT    = 000003
AC2    =%000002          CHRMNS= 000137    C.BLSZ= 000140    D.DECD= 000002    EIS     = 000000
AC3    =%000003          CHRMUL= 000075    C.BUFF= 177760    D.DIV = 000000    ELESIZ  005132R   025
AC4    =%000004          CHRNEG= 000100    C.BUFL= 177762    D.DMD = 000010    ENDCOR  000004R   025
AC5    =%000005          CHRPDL= 000271    C.CCNT= 177766    D.DQAD= 177777    EOF     = 104475
ALLINT  003274R   025     CHRPLU= 000055    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
ARYRET  003276R   025     CHRPWR= 000120    C.CZ  = 001000    D.ENCD= 000002    ERRNUM  004772R   025
ASCQUO= 000042            CHRQST= 000121    C.DIG = 000040    D.EQ  = 000000    ERRPTR  004770R   025
A.ALEN= 000004            CHRQUO= 000113    C.FASW= 020000    D.EXPD= 000002    ERRO    004774R   025
A.ARG = 000040            CHRRBR= 000047    C.FLEN= 000010    D.FGET= 000005    EXEBUF  005054R   025
A.ASB = 000200            CHRRPR= 000042    C.FLSH= 004000    D.FMT = 000004    E$CT    005056R   025
A.BITS= 000003            CHRSEM= 000074    C.FRC = 000001    D.FPNT= 000005    E$PT    005060R   025
A.CON = 000020            CHRSMO= 000112    C.IDEN= 000010    D.FPUT= 000005    FA.APD= 000100
A.DIMS= 000002            CHRSPA= 000040    C.NAME= 000102    D.GE  = 000000    FA.CRE= 000010
A.DIM1= 000006            CHRUPA= 000131    C.NOBF= 000400    D.GT  = 000000    FA.DLK= 001000
A.NUM = 000006            CHR$$ = 000101    C.NUM = 000020    D.IBM = 000002    FA.ENB= 100000
A.PTR = 000000            CHR.BS= 000010    C.OP  = 000004    D.IND = 000002    FA.EXC= 002000
A.TMP = 000010            CHR.CC= 000003    C.OPN = 010000    D.IMUR= 000013    FA.EXT= 000004
BADDIR= 104463            CHR.CR= 000015    C.RDLN= 177776    D.INTR= 000006    FA.NSP= 000100
BADERR= 104437            CHR.CZ= 000032    C.REWT= 002000    D.LE  = 000000    FA.NSR= 000200
BLZBUB  005116R   025     CHR.GT= 000046    C.RND = 000010    D.LOG = 000001    FA.POS= 010000
BRSELF  005046R   025     CHR.LF= 000012    C.SBIT= 177756    D.LT  = 000000    FA.RD = 000001
CCNT    005072R   025     CHR.LT= 000043    C.SPFX= 000100    D.MAX = 000000    FA.RWD= 004000
CHANEX  003060R   025     CHR.SI= 000017    C.SPGL= 000200    D.MEM = 000002    FA.SEQ= 040000
CHAN00  000256R   025     CHR.SL= 000057    C.SPLX= 000001    D.MIN = 000000    FA.SHR= 000040
CHAN01  000330R   025     CHR.SO= 000016    C.SPRS= 000002    D.MUL = 000000    FA.TMP= 000020
CHAN02  000512R   025     CHR.TB= 000011    C.SRC = 177774    D.NE  = 000000    FA.WCK= 020000
CHAN03  000674R   025     CHR.UL= 000106    C.SRCP= 000024    D.NN  = 000000    FA.WRT= 000002
CHAN04  001056R   025     CHR..A= 000141    C.TT  = 000004    D.NR  = 000000    FD.BLK= 000010
CHAN05  001240R   025     CHR..E= 000145    C.WIDC= 177770    D.OR  = 000000    FD.CCL= 000002
CHAN06  001422R   025     CHR..S= 000163    C.WIDR= 177772    D.PLUS= 000000    FD.COM= 020000
CHAN07  001604R   025     CHR..T= 000164    DEBUG = 000000    D.PSUB= 000006    FD.CR = 000002
CHAN08  001766R   025     CHR..U= 000165    DEFERR= 104405    D.PWR = 000001    FD.DIR= 000010
CHAN09  002150R   025     CHR..Z= 000172    DEFINE= 000000    D.QQAD= 000000    FD.FTN= 000001
CHAN10  002332R   025     CHR..0= 000060    DEVIL   005114R   025 D.QUAD= 000000    FD.F11= 040000
CHAN11  002514R   025     CHR..1= 000061    DEVNT?= 104470    D.RES = 000000    FD.INS= 000010
CHAN12  002676R   025     CH.AND= 000001    DIGITS  000006R   025 D.RHO = 000000    FD.ISP= 002000
CHAN13  000022RG  023     CMDBUF  003250R   025 DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN14  000204RG  023     CMDMAX= 000010    DIMBLK  003112R   025 D.SCON= 000006    FD.OSP= 004000
CHNOPN= 104471            CNTSIZ  005130R   025 DIMPTR  003114R   025 D.SDIF= 000006    FD.PLC= 000004
CHRAND= 000051            CNTSS = 000002    DIMTYP  003116R   025 D.SUB = 000000    FD.PRN= 000004
CHRBEL= 000007            COMBIN  005076R   025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRBKA= 000133            COMCNT  005126R   025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBOL= 000200            COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBOX= 000114            COMPTR  005122R   025 DYDCNT  003224R   025 D.UNON= 000006    FD.REC= 000001
CHRCMA= 000054            COMRTN  003156R   025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRCOL= 000076            COMSKP  005124R   025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRDEL= 000107            CORPTR  005120R   025 D.BXIU= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRDIV= 000053            CS.ASS= 000200    D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDLR= 000176            CS.BOO= 000010    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDLT= 000110            CS.FLO= 000020    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDOT= 000056            CS.IDN= 000040    D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHREP = 000105            CS.ONE= 000100    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRERR= 104431            CTEPS   000152R   025 D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRFIL= 000177            CTMAX   000212R   025 D.COMP= 000001    EDICHR  005063R   025 FF.RWF= 000006
CHRFRO= 000135            CTRMAX  000172R   025 D.CONT= 000006    EDICNT  005064R   025 FF.SPC= 000004
```

SYMBOL TABLE

```
FHEADR  005012R     025 F.FLB = 000100      HSHMSK= 177601          MONBLK  003120R     025 M.RDCT= 000011
FHRET   005040R     025 F.FNAM= 000110      HSHSIZ= 000200          MONCVT  003132R     025 M.REVS= 000003
FILNTF= 104467          F.FNB = 000102      ICHAN   005050R     025 MONLEN  003130R     025 M.ROLL= 000000
FIS   = 000000          F.FNDF= 000010      ILFNAM= 104464          MONPTR  003122R     025 M.SATA= 000001
FISCON= 046000          F.FTYP= 000116      ILLCHN= 104540          MONSFT  003126R     025 M.SCAN= 000011
FIXPTR  003074R     025 F.FUN = 000010      ILLSW = 104565          MONTYP  003124R     025 M.SHAP= 000000
FLAGS   004764R     025 F.FVER= 000120      INDERR= 104410          MONVAL  003134R     025 M.SIGN= 000000
FLCBUF  003250R     025 F.HEAD= 000006      INNTMP  003250R     025 M.ABS = 000000          M.SUBR= 000001
FLINE   005014R     025 F.HIBK= 000004      INNTM2  005016R     025 M.BOL = 000200          M.SYM = 000203
FLOEXP  003270R     025 F.LBL = 000020      INPTR   000064R     025 M.BXAV= 000007          M.SYN = 000201
FLONEG  003272R     025 F.LINE= 000012      INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FLSWTB  000222R     025 F.LOC = 000004      IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FLTLEN= 000010          F.LSTC= 000004      I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FNINSI  005044R     025 F.LUN = 000042      I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.APD= 000106          F.MBC1= 000055      I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.MFY= 0000C2          F.MBFG= 000056      I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.RD = 000001          F.MON = 000002      I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.UPD= 000006          F.NIL = 000001      I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FO.WRT= 000016          F.NRBD= 000024      LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FPMODE  000042R     025 F.NREC= 000030      LABLPT  005042R     025 M.BXRL= 000007          NAMBUF  003240R     025
FPP   = 000000          F.OPN = 000010      LARGBL  003162R     025 M.BXTS= 000007          NB.DEV= 000200
FPPLEN= 000004          F.OVBS= 000030      LARGCV  003176R     025 M.BXWA= 000007          NB.DIR= 000100
FREBLK  007400F     025 F.PREV= 000000      LARGFL  003160R     025 M.CEI = 177777          NB.NAM= 000004
FUZZ    000142R     025 F.PRNT= 100000      LARGLE  003174R     025 M.CHR = 000202          NB.SD1= 000400
FXPTR1  003076R     025 F.PROT= 010000      LARGLN  003166R     025 M.CMI = 000206          NB.SD2= 001000
F$EVAL= 000100          F.RACC= 000016      LARGPT  003164R     025 M.DOUT= 000006          NB.SNM= 000040
F$EVI = 000200          F.RATT= 000001      LARGSF  003172R     025 M.DQAD= 000006          NB.STP= 000020
F$EXEC= 000400          F.RCNM= 000034      LARGTY  003170R     025 M.ENCD= 000004          NB.SVR= 000010
F$FUNC= 001000          F.RCTL= 000017      LARGVA  003200R     025 M.EXE = 000000          NB.TYP= 000002
F.ACTL= 000076          F.READ= 004000      LBINC   005100R     025 M.EXP = 000001          NB.VER= 000001
F.ALOC= 000040          F.REDF= 000040      LBUFF   003316R     025 M.FACT= 000001          ND.DEV= 000200
F.BBFS= 000062          F.RES = 000040      LBUFLN= 000621          M.FIL = 000177          ND.EXT= 000002
F.BDB = 000070          F.RET = 177772      LCOUNT  005106R     025 M.FLR = 177777          ND.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002      LENERR= 104412          M.FMT = 000004          ND.PPN= 000100
F.BITS= 177774          F.RTYP= 000000      LENTAB  000054R     025 M.GDUP= 000003          ND.SEX= 000000
F.BKDN= 000026          F.R2  = 000014      LEXPTR  003100R     025 M.GDUN= 000003          ND.SES= 000020
F.BKDS= 000020          F.R3  = 000016      LFTOP   005066R     025 M.GETI= 000006          ND.SNM= 000000
F.BKEF= 000050          F.R4  = 000020      LINC    005102R     025 M.IBM = 000003          ND.SNS= 000040
F.BKP1= 000051          F.SEQN= 000100      LINE    005016R     025 M.IDN = 000000          ND.SPN= 001400
F.BKST= 000024          F.SPDV= 000072      LININC  005026R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BKVB= 000064          F.SPUN= 000074      LISIZE= 000020          M.IOTA= 000000          NINPT1  003302R     025
F.BR  = 000020          F.STBK= 000036      LISTDF= 000000          M.LIT = 000205          NINPT2  003300R     025
F.CHR = 000075          F.STK = 000022      LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.CNTG= 000034          F.STOP= 000004      LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CODE= 000010          F.STP = 000001      LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.DFNB= 000046          F.SUSP= 000002      LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
F.DSPT= 000044          F.TRC = 000002      LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.DVNM= 000134          F.UNIT= 000136      LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.EFBK= 000010          F.URBD= 000020          L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.EFN = 000052          F.VAL = 177776          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.FNU = 002000          F.VBN = 000064          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EOBB= 000032          F.VBSZ= 000060          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.ERR = 000053          GETL  = ****** GX       L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.FACC= 000043          GNOME   003102R     025 L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.FFBY= 000014          HASHC   005136R     025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000042R     025
F.FLAG= 000002          HASHTB  005140R     025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
```

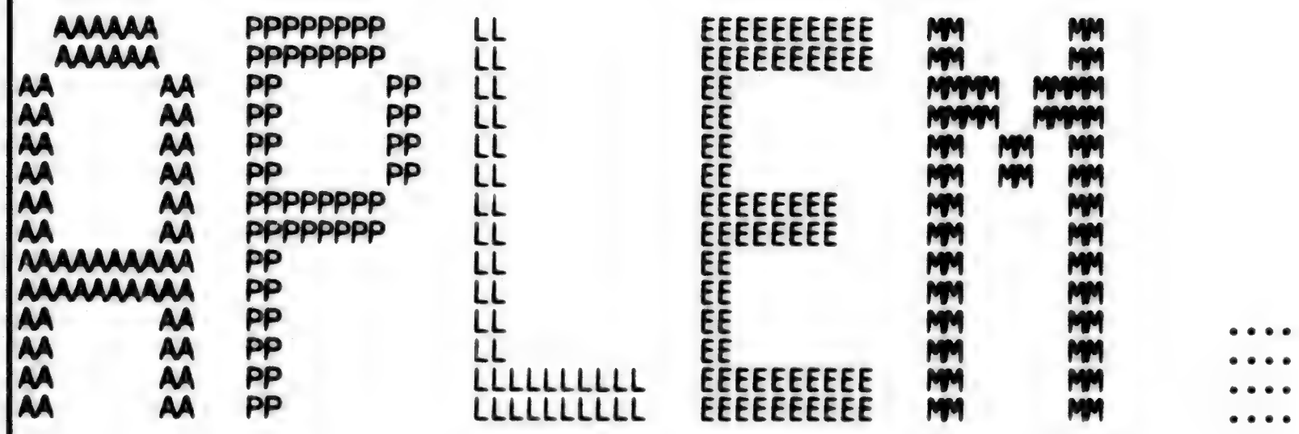
APLDF MACRO M1113 08-FEB-80 08:58 PAGE 45-3
SYMBOL TABLE

```
NUMQFL  003234R  025 RANDY   003062R  025 R.OFF  = 000000          SY.CLR= 000040          T.NIL  = 000004
NUMRIG  003231R  025 RANERR= 104411        R.PREV= 000004          SY.ERP= 000020          T.NUM  = 002000
NUMWID  003226R  025 RARGBL  003204R  025 R.PTR  = 000000          SY.NOU= 000100          T.OPFN= 000001
NUMZFL  003235R  025 RARGCV  003220R  025 R.SEQ  = 000003          SY.OPN= 000002          T.OTHR= 040000
NYIERR= 104432       RARGFL  003202R  025 R.VAR  = 000002          SY.PMT= 001000          T.SOP  = 000002
N.BLSZ= 000036       RARGLE  003216R  025 R6     =%000006          SY.SRT= 000400          T.SPOP= 000040
N.DESC= 000020       RARGLN  003210R  025 R7     =%000007          SY.SVC= 000004          T.TMP  = 000200
N.DEV  = 000032      RARGPT  003206R  025 SI     =%000005          SY.SVL= 000200          T.VAL  = 100000
N.DEVN= 000034       RARGSF  003214R  025 SIDAMN= 104433          S.AB   = 177776          T.VAR  = 001000
N.DID  = 000024      RARGTY  003212R  025 SIFRAM  003314R  025 S.ASB  = 100000          UPPER  = 177400
N.DVNM= 000032       RARGVA  003222R  025 SISIAK  006350R  025 S.BFHD= 000020          UTNAME  000344RG     023
N.EXT  = 000014      RBASE   C05074R  025 SITOP   006346R  025 S.BITS= 000002          VALERR= 104413
N.FID  = 000000      RCOUNR  005112R  025 SITP    007144R  025 S.CHR1= 000003          VARBLE  007400R  025
N.FNAM= 000006       RCOUNT  005110R  025 SIWARN= 000200          S.F..TT= 000016          VARBLS  000000R  025
N.FTYP= 000014       REDCNT  005070R  025 SIZTAB  000044R  025 S.FDB  = 000140          VERNUM  000000R  025
N.FVER= 000016       REDIGN  005066R  025 SIZ$$  = 000002          S.FLB  = 040000          VRSION= 000002
N.NAME= 000006       REDOFF  005072R  025 SI.BIT= 177774          S.FNAM= 000006          WIDTH   000072R  025
N.NEXT= 000022       REENTR  003060R  025 SI.LEN= 177772          S.FNB  = 000036          WROUND  000020R  025
N.PPN  = 000024      RESBIT  003142R  025 SI.RET= 177772          S.FNBW= 000017          WSDATE  000136R  025
N.STAT= 000020       RESCVT  003152R  025 SI.VAL= 177776          S.FNTY= 000004          WSFULL= 104403
N.UNIT= 000034       RESERR= 104440        SKPSIZ  005134R  025 S.FTYP= 000002          WSNAME  000074R  025
OBUFLN= 001000       RESLEN  003150R  025 SPHOLD  000004RG  024 S.LEN  = 000002          WSTIME  000132R  025
OCHAN   005052R  025 RESPTR  003140R  025 SPINIT  000002RG  024 S.LINK= 177774          $EIS   = 000000
OCHAR  = ****** GX   RESSFT  003146R  025 STKINC= 000774          S.NFEN= 000020          $FIS   = 000000 G
OMODE   000402RG 023 RESTYP  003144R  025 STOP   = 104420          S.TYPE= 140000          $FPP   = 000000 G
OPTMP0  005066R  025 RESULT  003136R  025 SUBCNT  003104R  025 S.VAL  = 000000          $FPPLN= 000004 G
OPTMP1  005070R  025 RESVAL  003154R  025 SUBDCT  003106R  025 TKDIMS  005072R  025 $MACHN= 000106 G
OPTMP2  005072R  025 RINC    005104R  025 SUBPTR  003110R  025 TKFRAM  005066R  025 $RSTS  = 000000 G
OPTMP3  005074R  025 RLTBAS  005066R  025 SYMB1   005344R  025 TKLEFT  005070R  025 $RSX   = 000115 G
OPTMP4  005076R  025 ROTINC  005070R  025 SYMF    005340R  025 TMODE   000403RG 023 $RT11  = 000000 G
OPTMP5  005100R  025 ROUND   000010R  025 SYMPT1  005346R  025 TMPTAG= 000166          $$     = 000001
OPTMP6  005102R  025 ROVER   003312R  025 SYMTAB  005352R  025 TTYBUF  004140R  025 .BOOLE= 000002
OPTMP7  005104R  025 RSTS   = 000000        SYMTBF= 104404        TTYLEN= 000622          .CHARA= 000000
OPTMP8  005106R  025 RSX    = 000115        SYMTOP  005350R  025 TURKEY= 104436          .DEF.  = 000002
OPTMP9  005110R  025 RTOP    005070R  025 SYMTP   006272R  025 TYPE   = 000166          .FLOAT= 000006
OPTM10  005112R  025 RT11   = 000000        SYMURN= 000054        T.BOL  = 020000          .FSRCB= ****** G
OPTM14  005026R  025 R.BITS= 000003        SYNERR= 104407        T.CON  = 004000          .INTEG= 000004
ORIGIN  000030R  025 R.FIX  = 000001        SYSERR= 104402        T.DIM  = 000100          .MOLUN  000404RG     023
OUTPTR  000066R  025 R.LCT  = 000002        SYSTAT  000000RG  024 T.FUN  = 010000          ...GBL= 000000
OVROPR= 000000       R.LN   = 000002        SY.BAD= 000001        T.LIST= 000400          ...PC1= 002676R  025
PATCH   007344R  025 R.LNUM= 000010        SY.BRK= 100000        T.LIT  = 000020          ...PC2= 003036R  025
PRECSN  000052R  025 R.NEXT= 000006        SY.BS  = 000010        T.MON  = 000010          ...TPC= 000020
RANDAC  003064R  025
```

```
. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
```

```
APLDF   MACRO M1113  08-FEB-80 08:58  PAGE 45-4
SYMBOL TABLE

TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000000      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13378 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:45
APLDX1,DM1:[305,301]APLDX1/-SP=APLRX1,APLDF,APLDEF,END
```

FILEID**APLEM



.MAIN. MACRO M1113 08-FEB-80 09:02
TABLE OF CONTENTS


```
     2-     1     APLDEF
     2-     8     A P L
     2-     9
     2-    10     A PROGRAMMING LANGUAGE
     2-    11
     2-    12     STEVE POULSEN & DAVE VANN
     2-    13     11-Jul-76
     2-    14     ROBERT GOTTLIEB & PETER HAYNES
     2-    15     2-Feb-79
     2-    16
     2-    17     V2
     2-    18
     2-    19
     2-    20     COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25     COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26     COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27     BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28     USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29     TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30     DEC.
     2-    31
     2-    32     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33     AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34     CORPORATION.
     2-    35
     2-    36     DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37     SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46     DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139     DEFINITIONS      OPERATOR OVERLAYS
     5-   268     DEFINITIONS      REGISTERS
     6-   331     DEFINITIONS      MEMORY ORGANIZATION
     7-   377     DEFINITIONS      ORDERING PSECTS
     8-   426     DEFINITIONS      SI PARSE FRAMES
     9-   474     DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515     DEFINITIONS      SI EXECUTION FRAMES
    11-   580     DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607     DEFINITIONS      CHRTAB BITS
    12-   624     DEFINITIONS      SYSTEM STATUS BITS
    13-   640     DEFINITIONS      F-FRAMES
    14-   724     DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807     DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869     DEFINITIONS      LIST STORAGE BLOCK
    18-   911     DEFINITIONS      INDEXING STACK FRAME
    19-   940     DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969     DEFINITIONS      CHANNEL BLOCK
    22-  1085     DEFINITIONS            RSX DATA
    23-  1110     DEFINITIONS      NAME BLOCKS
    33-  1612     DEFINITIONS      MACROS
    35-  1905     DEFINITIONS      APL CHARACTERS
    36-  1967     DEFINITIONS      ERRORS
    44-     5     ISETUP           SET UP FOR FAKE REG INTRSUCTIONS
```

APLEM MACRO M1113 08-FEB-80 09:02
TABLE OF CONTENTS

46- 91 IXOR 'XOR' SIMULATOR
46- 103 IASH 'ASH' SIMULATOR
47- 121 IASHC 'ASHC' SIMULATOR
48- 176 IMUL 'MUL' SIMULATOR
49- 220 IDIV 'DIV' SIMULATOR
50- 295 FPCODE FAKE THE HARDWARE FPP

```
APLEM   MACRO M1113  08-FEB-80 09:02  PAGE 1


      1  000000                RT11    =  0          ;NOT RT-11
      2  000000                RSTS    =  0          ;NOT RSTS/E
      3  000115                RSX     =  'M         ;RSX-11/M
      4
      5  000000                EIS     =  0          ;NO EIS
      6  000000                FIS     =  0          ;NO FIS
      7  000000                FPP     =  0          ;NO FPP
      8
      9  000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11  000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  APLEM
                                .IDENT  \V02.03\

                                .SBTTL  ISETUP          SET UP FOR FAKE REG INTRSUCTIONS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       SETS UP FOR SUCH SIMULATIONS AS XOR, MUL, DIV, ASH, AND ASHC.
                                ;       THIS IS CALLED BY THE SEPARATE SIMULATION ROUTINES TO SET UP
                                ;       POINTERS AND SAVE REGISTERS AND EXIT.  WHEN CALLED THE SP STACK
                                ;       LOOKS LIKE:
                                ;
                                ;       ******************
                                ;       * SOURCE VALUE  *        VALUE OF LEFT ARG IN THE INSTRUCTION
                                ;       ******************
                                ;       *    REG #       *        REGISTER TO USE (MINUS IF ONLY 1 REGISTER
                                ;       ******************        ELSE USES 'R' AND 'R OR 1')
                                ;       *    RETURN      *        RETURN POINTER TO CALLING 'INSTRUCTION'
                                ;       ******************
                                ;       *    RETURN      *        RETURN POINTER TO SIMULATION ROUTINE
                                ;       ******************
                                ;
                                ;       THIS ROUTINE SETS ALL FLAGS TO ZERO FOR INDIVIDUAL ROUTINES TO SET
                                ;       WHEN NEEDED.
                                ;
                                ; INPUTS:
                                ;       R1 -    SOURCE VALUE
                                ;       R2 -    REGISTER VALUE
                                ;       R3 -    (REGISTER OR 1) VALUE
                                ;--

                                ;+
                                ; VARIABLES FOR XOR, MUL, DIV, ASH, ASHC SIMULATOR
                                ;-
000000                                  .PSECT  EMULAT, RW, D, LCL

000000  000000         IFLAGS: .WORD   0               ;CONDITION BITS
000002  000000         IREG0:  .WORD   0               ;POINTER TO FIRST REG
000004  000000         IREG1:  .WORD   0               ;POINTER TO (REG OR 1)
000006  000000         IRETRN: .WORD   0               ;POINTER TO ROUTINE TO USE
000010  000000         IREG:   .WORD   0               ;REGISTER NUMBER BEING USED
000012  000000         ISRC:   .WORD   0               ;SOURCE VALUE
000014  000000         IRETR1: .WORD   0               ;RETURN POINTER

                000001          IC=01                   ;CONDITION CODE DEFINITIONS
                000002          IV=02
                000004          IZ=04
                000010          IN=10
```

```
APLEM   MACRO M1113  08-FEB-80 09:02  PAGE 45
ISETUP          SET UP FOR FAKE REG INTRSUCTIONS

     56 000000                                          .PSECT  CODE
     57
     58 000000  005037  000000'          ISETUP: CLR     IFLAGS          ;ZERO CONDITIONAL FLAGS
     59 000004                                  POP     <IRETRN,IRETR1,IREG,ISRC>;GET SOME STUFF OFF THE STACK
     60 000024                                  PUSH    <R4,R3,R2,R1,R0>;AND SAVE SOME REGISTERS ON IT
     61 000036  013701  000012'                 MOV     ISRC,R1         ;GET SOURCE VALUE
     62 000042  013704  000010'                 MOV     IREG,R4         ;GET REGISTER TO USE
     63 000046  006304                          ASL     R4              ;WORD OFFSET ON STACK
     64 000050  100414                          BMI     1$              ;ONLY ONE REGISTER IF NEGATIVE
     65
     66 000052  010400                          MOV     R4,R0           ;COPY REG #
     67 000054  060604                          ADD     SP,R4           ;POINT TO REG VALUE
     68 000056  011402                          MOV     (R4),R2         ;GET REG VALUE
     69 000060  010437  000002'                 MOV     R4,IREG0        ;SAVE POINTER
     70 000064  052700  000002                  BIS     #2,R0           ;REGISTER OR 1
     71 000070  060600                          ADD     SP,R0           ;POINT TO IT
     72 000072  011003                          MOV     (R0),R3         ;AND GET IT
     73 000074  010037  000004'                 MOV     R0,IREG1        ;SAVE POINTER FOR LATER
     74 000100  000405                          BR      2$              ;DO INDIVIDUAL ROUTINE
     75
     76 000102  005404                  1$:     NEG     R4              ;MAKE POSITIVE
     77 000104  060604                          ADD     SP,R4           ;POINT TO REGISTER
     78 000106  011402                          MOV     (R4),R2         ;GET REG VALUE
     79 000110  010437  000002'                 MOV     R4,IREG0        ;SAVE POINTER
     80
     81 000114                          2$:     CALL    @IRETRN         ;DO ROUTINE
     82 000120  010277  000002'                 MOV     R2,@IREG0       ;SAVE 1 REG
     83 000124  105737  000010'                 TSTB    IREG            ;1 OR 2 REGS?
     84 000130  100402                          BMI     3$              ;ONLY 1!
     85 000132  010377  000004'                 MOV     R3,@IREG1       ;OR ONE IF ODD REGISTER
     86
     87 000136                          3$:     POP     <R0,R1,R2,R3,R4>;RESTORE REGISTERS
     88 000150                                  PUSH    <IFLAGS,IRETR1> ;SET FOR RTI TYPE RETURN
     89 000160  000002                          RTI                     ;SET PROPER CONDITION CODES
```

```
                                .SBTTL  IXOR            'XOR' SIMULATOR

                                .PSECT CODE

000162                  IXOR::  CALL    ISETUP          ;SET UP VALUES
000166  010200                 MOV     R2,R0           ;COPY SOURCE
000170  050102                 BIS     R1,R2           ;OR SORCE AND REGISTER
000172  005101                 COM     R1              ;LOGICAL NOT ON REGISTER
000174  040100                 BIC     R1,R0           ;CLEAR A FEW BITS
000176  040002                 BIC     R0,R2           ;AND A FEW MORE
000200  000463                 BR      INZBIT          ;AND ZAPPO, WE'RE DONE

                                .SBTTL  IASH            'ASH' SIMULATOR

000202                  IASH::  CALL    ISETUP          ;SET UP VALUES
000206  010204                 MOV     R2,R4           ;SAVE REGISTER VALUE
000210  005701                 TST     R1              ;WHICH WAY?
000212  001430                 BEQ     IASHX           ;NIETHER! DONE REAL FAST
000214  100004                 BPL     2$              ;LEFT!

000216  006202          1$:    ASR     R2              ;SHIFT ONCE
000220  005201                 INC     R1              ;DONE YET?
000222  001375                 BNE     1$              ;NOPE
000224  000423                 BR      IASHX           ;YES, SET CONDITION CODES

000226  006302          2$:    ASL     R2              ;LEFT ONCE
000230  005301                 DEC     R1              ;DONE YET?
000232  001375                 BNE     2$              ;NOPE
000234  000417                 BR      IASHX           ;YES, SET CONDITION CODES
```

```
                                .SBTTL  IASHC           'ASHC' SIMULATOR

000236                          .PSECT CODE

000236                  IASHC:: CALL    ISETUP          ;SET UP SOME VALUES
000242  010204                  MOV     R2,R4           ;SAVE REGISTER VALUE
000244  005701                  TST     R1              ;WHICH WAY?
000246  001412                  BEQ     IASHX           ;NEITHER! DONE
000250  100005                  BPL     2$              ;GO LEFT YOUNG MAN

000252  006202          1$:     ASR     R2              ;SHIFT HIGH RIGHT
000254  006003                  ROR     R3              ;SHIFT LOW RIGHT WITH CARRY
000256  005201                  INC     R1              ;DONE?
000260  001374                  BNE     1$              ;NO
000262  000404                  BR      IASHX           ;YUP, WE IS, SHO 'NUFF

000264  006303          2$:     ASL     R3              ;SHIFT LOW LEFT
000266  006102                  ROL     R2              ;SHIFT HIGH LEFT WITH CARRY
000270  005301                  DEC     R1              ;DONE YET?
000272  001374                  BNE     2$              ;NO WAY

                        ;SET 'C' AND 'V' BITS FOR 'ASHC' AND 'ASH'

000274  103003          IASHX:  BCC     1$              ;CHECK 'C' BIT
000276  052737  000001  000000' BIS     #IC,IFLAGS      ;SET 'C' IF CARRY SET
000304  010400          1$:     MOV     R4,R0           ;CHECK 'V'
000306  010201                  MOV     R2,R1           ;XOR SIGNS OF ORIGINAL REG AND RESULT
000310  050104                  BIS     R1,R4           ;IF NO ASH THEN NO XOR SO DO IT THE
000312  005101                  COM     R1              ;    LONG WAY.
000314  040100                  BIC     R1,R0           ;DRAWN OUT AND BORING
000316  040004                  BIC     R0,R4           ;SIMILAR TO THE 'XOR' ROUTINE
000320  100003                  BPL     INZBI2          ;NOT SET IF SAME SIGN
000322  052737  000002  000000' BIS     #IV,IFLAGS      ;SIGN CHANGE, SET 'V'

                        ;SET 'N' AND 'Z' ACCORDING TO R2 AND R3

000330  005703          INZBI2: TST     R3              ;LOW ZERO?
000332  001013                  BNE     INZBI1          ;NO
000334  005702                  TST     R2              ;HIGH ZERO?
000336  001011                  BNE     INZBI1          ;NO
000340  052737  000004  000000' BIS     #IZ,IFLAGS      ;SET 'Z' BIT
000346  000405                  BR      INZBI1          ;CHECK ON 'N' BIT

                        ;SET 'N' AND 'Z' ACCORDING TO R2

000350  005702          INZBIT: TST     R2              ;R2 ZERO?
000352  001003                  BNE     INZBI1          ;NO
000354  052737  000004  000000' BIS     #IZ,IFLAGS      ;YES, SET 'Z'

000362  005702          INZBI1: TST     R2              ;NEGATIVE?
000364  100003                  BPL     1$              ;NO
000366  052737  000010  000000' BIS     #IN,IFLAGS      ;YES, SHOW WE'RE NEGATIVE

000374                  1$:     RETURN                  ;DONE!!!
```

```
                        .SBTTL  IMUL            'MUL' SIMULATOR

                        .PSECT CODE

000376                  IMUL::  CALL    ISETUP          ;GET SOME STUFF PREPARED FOR US
000402  005701                  TST     R1              ;POSITIVE SOURCE?
000404  100002                  BPL     1$              ;YES
000406  005401                  NEG     R1              ;MAKE SURE AT LEAST ONE IS POSITIVE
000410  005402                  NEG     R2              ;PRESERVE RESULT SIGN
000412  010200          1$:     MOV     R2,R0           ;COPY REG VALUE
000414  020001                  CMP     R0,R1           ;IS SOURCE LOWER?
000416  101402                  BLOS    2$              ;NO
000420  010100                  MOV     R1,R0           ;LOWEST IN R0
000422  010201                  MOV     R2,R1           ;HIGHEST IN R1
000424  005002          2$:     CLR     R2              ;CLEAR RESULT
000426  005003                  CLR     R3              ;DOUBLE PRECISION RESULT
000430  005004                  CLR     R4              ;MAKE R4 HIGH PART OF R1
000432  005701                  TST     R1              ;WHAT SIGN FOR SIGN EXTENSION
000434  100001                  BPL     3$              ;POSITIVE, LEAVE ZERO
000436  005104                  COM     R4              ;EXTEND SIGN NOT ON THIS MACHINE

                        ;LOOP AND MULTIPLY BY SHIFTS AND ADDS

000440  006200          3$:     ASR     R0              ;SHIFT MULTIPLIER RIGHT
000442  103414                  BCS     7$              ;ADD THIS TIME
000444  001016                  BNE     8$              ;NO ADD

000446  005702                  TST     R2              ;SET C-BIT, IF SIGNS OF HIGH HALF
000450  100006                  BPL     6$              ;  AND LOW HALF
000452  005703                  TST     R3              ;  DON'T MATCH
000454  100403                  BMI     5$              ;THEY DO!
000456  052737  000001  000000' 4$:     BIS     #IC,IFLAGS      ;SET 'C' FOR MORE THEN +-2^15
000464  000721          5$:     BR      INZBI2          ;GO TAKE CARE OF 'Z' AND 'N' BITS
000466  005703          6$:     TST     R3              ;IS LOW HALF NEGATIVE?
000470  100772                  BMI     4$              ;YES
000472  000774                  BR      5$              ;NO

000474  060103          7$:     ADD     R1,R3           ;ADD TO RESULT
000476  005502                  ADC     R2              ;DON'T FORGET OVERFLOW OF ADD
000500  060402                  ADD     R4,R2           ;WE'RE DOING DOUBLE PRECISION
000502  006301          8$:     ASL     R1              ;SHIFT ONE MORE LEFT
000504  006104                  ROL     R4              ;DOUBLE PRECISION
000506  000754                  BR      3$              ;GO DO IT AGAIN SAM
```

```
                                .SBTTL  IDIV            'DIV' SIMULATOR

000510                          .PSECT CODE

000510                  IDIV::  CALL    ISETUP          ;SET UP SOME VALUES
000514  005000                  CLR     R0              ;USE [R0:R1] AS A DOUBLE REGISTER
000516  005004                  CLR     R4              ;ZERO QUOTIENT
000520                          PUSH    R2              ;SAVE SIGN OF DIVIDEND
000522  005701                  TST     R1              ;DIVISOR NEGATIVE?
000524  001467                  BEQ     11$             ;MOMMY! HELP! DIVISION BY ZERO!!
000526  100004                  BPL     2$              ;NOPE
000530  005401                  NEG     R1              ;MAKE POSITIVE
000532  005403                  NEG     R3              ;NEGATE DIVIDEND TOO
000534  005502                  ADC     R2              ;DOUBLE PRECISION NEGATE
000536  005402                  NEG     R2              ;A REAL PAIN
000540  010137  000012'    2$:   MOV     R1,ISRC         ;REPLACE POSITIVE SOURCE
000544                          PUSH    R2              ;SAVE SIGN OF RESULT
000546  100003                  BPL     3$              ;IS DIVIDEND NOW POSITIVE
000550  005403                  NEG     R3              ;NO, MAKE POSITIVE
000552  005502                  ADC     R2              ;DOUBLE PRECISION NEGATE
000554  005402                  NEG     R2              ;REAL TRICKY NEGATE
000556  006301          3$:     ASL     R1              ;LEFT JUSTIFY SOURCE
000560  100376                  BPL     3$              ;THERE YET?
000562  006001                  ROR     R1              ;YES! MOVE BACK RIGHT ONE

                                ;DO THE GRUNGY DIVISION

000564  160003          4$:     SUB     R0,R3           ;TAKE DIVISOR FROM DIVIDEND
000566  005602                  SBC     R2              ;DOUBLE PRECISION SUBTRACT
000570  160102                  SUB     R1,R2           ;DID WE TAKE OFF TOO MUCH?
000572  100004                  BPL     5$              ;NOPE
000574  060003                  ADD     R0,R3           ;PUT IT BACK ON
000576  005502                  ADC     R2              ;ALL OF IT
000600  060102                  ADD     R1,R2           ;THE WHOLE DOUBLE PRECISION REG
000602  000401                  BR      6$              ;SHIFT DIVISOR RIGHT 1

000604  005204          5$:     INC     R4              ;SET BIT IN QUOTIENT
000606  006201          6$:     ASR     R1              ;SHIFT DIVISOR RIGHT
000610  006000                  ROR     R0              ;DOUBLE PRECISION RIGHT
000612  005701                  TST     R1              ;HIGH NON-ZERO?
000614  001003                  BNE     12$             ;YES, NOT DONE YET!
000616  020037  000012'         CMP     R0,ISRC         ;DONE YET
000622  103413                  BLO     7$              ;YES
000624  006304          12$:    ASL     R4              ;BUMP QUOTIENT
000626  100356                  BPL     4$              ;GOOD IF NO OVERFLOW
000630  017702  000002'    8$:   MOV     @IREG0,R2       ;OVERFLOW, RESTORE REGISTERS
000634  017703  000004'         MOV     @IREG1,R3       ;BOTH OF THEM
000640  052737  000002  000000' BIS     #IV,IFLAGS      ;OVERFLOW FLAG!
000646  022626                  CMP     (SP)+,(SP)+     ;CLEAN STACK
000650                          RETURN

                                ;FINISH UP DIVISION

000652  005702          7$:     TST     R2              ;REMAINDER TOO BIG??
000654  001365                  BNE     8$              ;YES! BAD!
000656  005703                  TST     R3              ;ONLY A LITTLE TOO BIG?
000660  100763                  BMI     8$              ;YAS! HA-HA
```

```
    277 000662  005726                          TST     (SP)+           ;SHOULD QUOTIENT BE NEGATIVE?
    278 000664  100001                          BPL     9$              ;NO
    279 000666  005404                          NEG     R4              ;NEGATE QUOTIENT
    280 000670  005726                  9$:     TST     (SP)+           ;AND WHAT ABOUT THE SIGN OF REMAINDER?
    281 000672  100001                          BPL     10$             ;LEAVE POSITIVE
    282 000674  005403                          NEG     R3              ;NEGATE IT PLEASE
    283 000676  010402                  10$:    MOV     R4,R2           ;MOVE QUOTIENT TO PROPER PLACE
    284 000700  000137  000350'                 JMP     INZBIT          ;CHECK ON 'Z' AND 'N' BITS
    285
    286                                 ;DIVIDE BY ZERO ERROR! HE DID A NO-NO!!
    287
    288 000704  005726                  11$:    TST     (SP)+           ;CLEAN STACK
    289 000706  052737  000003  000000'          BIS     #IV!IC,IFLAGS   ;SET 'C' AND 'V' FLAGS
    290 000714                                  RETURN                  ;BACK TO COMMON ROUTINE
    291
```

```
                                        .SBTTL FPCODE           FAKE THE HARDWARE FPP

                                ;+
                                ; Variables for PDP-11 FPP Simulator
                                ;-
                                        .PSECT  EMULAT, RW, D, LCL

000016                          FVALA:  .BLKW   0               ;VALUE 'A'
000016                          FSGNA:  .BLKW   1               ;SIGN OF A
000020                          FMANA:  .BLKW   4               ;MANTISSA OF A
000030                          FEXPA:  .BLKW   1               ;EXPONENT OF A
000032                          FVALB:  .BLKW   0               ;VALUE B
000032                          FSGNB:  .BLKW   1               ;SIGN OF B
000034                          FMANB:  .BLKW   4               ;MANTISSA OF B
000044                          FEXPB:  .BLKW   1               ;EXPONENT OF B
000046  000000                  FSRC:   .WORD   0               ;SOURCE POINTER
000050  000016'                         .WORD   FVALA
000052  000000                  FDST:   .WORD   0               ;DESTINATION POINTER
000054  000032'                         .WORD   FVALB
000056                          FSTAT:  .BLKW   1               ;FPP STATUS WORD
000060                          FSTATS: .BLKW   1               ;SAVED STATUS WORD
000062                          FTMP:   .BLKW   6.              ;FPP'S TEMP LOCATIONS
000076                          REG0:   .BLKW   1               ;SAVED R0
000100                          REG1:   .BLKW   1               ;       R1
000102                          REG2:   .BLKW   1               ;       R2
000104                          REG3:   .BLKW   1               ;       R3
000106                          REG4:   .BLKW   1               ;       R4
000110                          REG5:   .BLKW   1               ;       R5
000112                          REG6:   .BLKW   1               ;       SP
000114                          REG7:   .BLKW   1               ;       PC
000116                          FMOV:   .BLKW   1               ;PSEUDO MOVE INSTRUNCTION
000120                          FDESC:  .BLKW   1               ;SRC/DST DESC BITS FOR OP
000122                          FRTN:   .BLKW   1               ;RETURN POINTER FOR 'FSAVIT' ROUTINE
000124                          FFLAGS: .BLKW   1               ;CONDITION FLAGS TO SET ON RETURN
000126                          FSIZE:  .BLKW   1               ;SIZE OF FPP #'S
000130                          FISIZE: .BLKW   1               ;SIZE IF FPP'S INTEGERS
000132                          FAC0::  .BLKW   FLTLEN/2
000142                          FAC1::  .BLKW   FLTLEN/2
000152                          FAC2::  .BLKW   FLTLEN/2
000162                          FAC3::  .BLKW   FLTLEN/2
000172                          FAC4::  .BLKW   FLTLEN/2
000202                          FAC5::  .BLKW   FLTLEN/2
```

```
                                        .PSECT CODE

                                ;+
                                ;BITS USED TO DESCRIBE OPERATION TO 'FSAVIT' AND 'FDONE' ROUTINES
                                ;-
000001                          ..SRC   =       000001  ;SOURCE POINTER
000002                          ..ISRC  =       000002  ;INTEGER SOURCE
000004                          ..FSRC  =       000004  ;FLOATING SOURCE
000010                          ..GSRC  =       000010  ;GET SOURCE TO VALUE 'A'
000400                          ..DST   =       000400  ;DESTINATION POINTER (UPPER BYTE!)
001000                          ..IDST  =       001000  ;INTEGER DESTINATION
002000                          ..FDST  =       002000  ;FLOATING DESTINATION
004000                          ..GDST  =       004000  ;GET DESTINATION TO VALUE 'B'
010000                          ..PDST  =       010000  ;SAVE FLOATING DESTINATION
020000                          ..JDST  =       020000  ;SAVE INTEGER DESTINATION
100000                          ..CFLG  =       100000  ;CHECK FLAGS

                                ;+
                                ;BITS OF FPP STATUS WORD, 'FSTAT'
                                ;-
000200                          FS.DBL  =          200  ;DOUBLE PRECISION
000100                          FS.LNG  =          100  ;LONG INTEGER
000010                          FS.N    =           10  ;N BIT
000004                          FS.Z    =            4  ;Z BIT
000002                          FS.V    =            2  ;V BIT
000001                          FS.C    =            1  ;C BIT

                                ;+
                                ;SAVE REGISTERS AND GET SRC/DST POINTERS/VALUES IF NEEDED
                                ;-
000716  010037  000114'         FSAVIT: MOV     R0,REG7         ;SAVE IT AGAIN
000722  011637  000120'                 MOV     (SP),FDESC      ;SAVE POINTER TO BITS DESCRIBING ARGS
000726  062716  000002                 ADD     #2,(SP)         ;POINT TO RETURN TO FPP ROUTINE
000732                                  POP     <FRTN,REG0>     ;SAVE RETURN POINTER AND R0
000742  012700  000100'                 MOV     #REG1,R0        ;POINT TO WHERE TO SAVE REGISTERS
000746  010120                         MOV     R1,(R0)+        ;SAVE R1
000750  010220                         MOV     R2,(R0)+        ;    AND R2
000752  010320                         MOV     R3,(R0)+        ;        AND R3
000754  010420                         MOV     R4,(R0)+        ;            AND R4
000756  010520                         MOV     R5,(R0)+        ;                AND R5
000760  010620                         MOV     SP,(R0)+        ;                    AND R6
000762  017705  000120'                 MOV     @FDESC,R5       ;GET BITS DESCRIBING ARGS
000766  010537  000120'                 MOV     R5,FDESC        ;SAVE FLAGS AGAIN
000772  012737  000015  000124'         MOV     #15,FFLAGS      ;SAVE FPP CONDITION FLAGS
001000  032705  100000                 BIT     #..CFLG,R5      ;CHECK FLAGS?
001004  001403                         BEQ     3$              ;NO
001006  042737  000017  000056'         BIC     #17,FSTAT       ;CLEAR FLAGS TO START WITH

001014  013704  000114'         3$:     MOV     REG7,R4         ;GET INSTR PC
001020  162706  000020                 SUB     #20,SP          ;GET SOME BREATHING ROOM ON THE STACK
001024  032705  000401                 BIT     #..SRC!..DST,R5 ;ARGS TO PROCESS?
001030  001430                         BEQ     2$              ;NOPE
001032  012437  000116'                 MOV     (R4)+,FMOV      ;GET DESCRIPTIVE 'MOV' INSTR
001036  032705  000001                 BIT     #..SRC,R5       ;DECODE SOURCE ARG?
001042  001411                         BEQ     1$              ;NO
001044  013703  000116'                 MOV     FMOV,R3         ;GRAB MOV INSTR
001050  006303                         ASL     R3              ;SRC BITS TO RIGHT
```

```
APLEM   MACRO M1113  08-FEB-80 09:02  PAGE 51-1
FPCODE           FAKE THE HARDWARE FPP

     403 001052  006303                          ASL R3
     404 001054  000303                          SWAB R3
     406 001056  012702  000046'                 MOV     #FSRC,R2        ;POINT TO WHERE TO PUT ARG INFO
     407 001062                                  CALL    FPTR            ;GET POINTER AND DECODE IF NECESSARY
     408
     409 001066  032705  000400          1$:     BIT     #..DST,R5       ;DECODE DST BITS?
     410 001072  001407                          BEQ     2$              ;NOPE!
     411 001074  000305                          SWAB    R5              ;SHIFT DST BITS TO LOWER BYTE
     412 001076  013703  000116'                 MOV     FMOV,R3         ;GET DST DESC BITS
     413 001102  012702  000052'                 MOV     #FDST,R2        ;POINTER TO WHERE TO PUT ARG INFO
     414 001106                                  CALL    FPTR            ;GET POINTER AND ARG IF NESCESSARY
     415
     416 001112  010437  000114'          2$:    MOV     R4,REG7         ;SAVE NEW PC
     417 001116  000177  000122'                 JMP     @FRTN           ;RETURN TO CALLING FPP ROUTINE
     418
     419
```

```
                                        ;
                                        ;DECODE POINTER DESCRIPTION BITS AS IN 'MOV' INSTRUCTION
                                        ;
001122  010437  000114'         FPTR:  MOV     R4,REG7          ;SAVE UPDATED PC
001126  010301                         MOV     R3,R1            ;COPY PTR BITS
001130  042701  177770                 BIC     #^C7,R1          ;MASK TO REG BITS
001134  006301                         ASL     R1               ;MULTIPLY BY 2 FOR WORDS
001136  062701  000076'                ADD     #REG0,R1         ;POINT TO REG TO USE
001142  006203                         ASR     R3               ;SHIFT MODE BITS TO RIGHT * 2
001144  006203                         ASR R3
001146  042703  177761                 BIC     #^C16,R3         ;MASK TO MODE BITS
001152  032705  000004                 BIT     #..FSRC,R5       ;FLOATING ARG?
001156  000173  001162'                JMP     @FMTBL(R3)       ;GOTO PROPER ROUTINE FOR MODE

001162  001202'                FMTBL: FMODE0
001164  001226'                       FMODE1
001166  001234'                       FMODE2
001170  001320'                       FMODE3
001172  001332'                       FMODE4
001174  001372'                       FMODE5
001176  001404'                       FMODE6
001200  001420'                       FMODE7

001202  001406                 FMODE0: BEQ     1$               ;AC, NOT REG
001204  162701  000076'                SUB     #REG0,R1         ;GET REG # * 2
001210  006301                         ASL     R1               ;*4
001212  006301                         ASL     R1               ;*10
001214  062701  000132'                ADD     #FAC0,R1         ;POINT TO AC

001220  010100                 1$:    MOV     R1,R0            ;COPY POINTER
001222  005003                         CLR     R3               ;NO REG INC/DEC
001224  000503                         BR      FGETIT           ;FIX UP POINTER

001226  011100                 FMODE1: MOV     (R1),R0          ;DEFER OFF REG FOR PTR
001230  005003                         CLR     R3               ;NO REG INC/DEC
001232  000500                         BR      FGETIT           ;FIX UP PTR

001234  001417                 FMODE2: BEQ     1$               ;REG, NOT AC
001236  013703  000126'                MOV     FSIZE,R3         ;GET FLOATING INC
001242  020127  000114'                CMP     R1,#REG7         ;IMMEDIATE CONST?
001246  001022                         BNE     3$               ;NO
001250  012703  000002                 MOV     #2,R3            ;INC ONLY 2 THEN (SPECIAL CASE)
001254  012700  000072'                MOV     #FTMP+10,R0      ;POINT TO WHERE TO FAKE OUT ARG
001260  005040                         CLR     -(R0)            ;CLR LSB
001262  005040                         CLR -(R0)
001264  005040                         CLR -(R0)
001266  017140  000000                 MOV     @(R1),-(R0)      ;COPY MSB AND MAKE POINTER
001272  000460                         BR FGETIT

001274  032705  000002         1$:    BIT     #..ISRC,R5       ;INTEGER SOURCE?
001300  001403                         BEQ     2$               ;NOPE
001302  013703  000130'                MOV     FISIZE,R3        ;GET CURRENT INTEGER SIZE
001306  000402                         BR 3$

001310  012703  000002         2$:    MOV     #2,R3            ;GET WORD SIZE
```

FPCODE FAKE THE HARDWARE FPP

```
                                                
001314  011100                  3$:     MOV     (R1),R0         ;GET VALUE OF REGISTER
001316  000446                          BR FGETIT
                                        
001320  012703  000002          FMODE3: MOV     #2,R3           ;INC REG BY 2
001324  017100  000000                  MOV     @(R1),R0        ;REG IS PTR TO PTR
001330  000441                          BR FGETIT
                                        
001332  001011                  FMODE4: BNE     1$              ;F-ARG
001334  032705  000002                  BIT     #..ISRC,R5      ;INTEGER SOURCE?
001340  001403                          BEQ     3$              ;NO
001342  013703  000130'                  MOV     FISIZE,R3       ;GET INTEGER SIZE
001346  000405                          BR 2$
                                        
001350  012703  000002          3$:     MOV     #2,R3           ;WORD SOURCE
001354  000402                          BR 2$
                                        
001356  013703  000126'          1$:     MOV     FSIZE,R3        ;FLOATING SIZE
                                        
001362  005403                  2$:     NEG     R3              ;AUTO DEC THIS TIME
001364  011100                          MOV     (R1),R0         ;GET POINTER PAST ARG
001366  060300                          ADD     R3,R0           ;POINT TO ARG
001370  000421                          BR FGETIT
                                        
001372  012703  177776          FMODE5: MOV     #-2,R3          ;DEC REG BY 2
001376  011100                          MOV     (R1),R0         ;GET PTR PAST PTR
001400  014000                          MOV     -(R0),R0        ;MAKE PTR TO ARG
001402  000414                          BR FGETIT
                                        
001404  005003                  FMODE6: CLR     R3              ;NO INC/DEC
001406  012400                          MOV     (R4)+,R0        ;GET INDEX CONST
001410  010437  000114'                  MOV     R4,REG7         ;UPDATE PC
001414  061100                          ADD     (R1),R0         ;ADD IN REG VALUE
001416  000406                          BR FGETIT
                                        
001420  005003                  FMODE7: CLR     R3              ;NO INC/DEC
001422  012400                          MOV     (R4)+,R0        ;GET INDEX CONST
001424  010437  000114'                  MOV     R4,REG7         ;UPDATE PC
001430  061100                          ADD     (R1),R0         ;ADD REG VALUE
001432  011000                          MOV     (R0),R0         ;DEFER OFF IT!
                                        
001434  010437  000114'          FGETIT: MOV     R4,REG7         ;UPDATE PC
001440  060311                          ADD     R3,(R1)         ;INC REG (IF NEEDED)
001442  013704  000114'                  MOV     REG7,R4         ;IN CASE PC CHANGED
001446  010022                          MOV     R0,(R2)+        ;SAVE PTR
001450  032705  000010                  BIT     #..GSRC,R5      ;GET FLOATING ARG?
001454  001425                          BEQ     2$              ;NO
001456  011202                          MOV     (R2),R2         ;GET POINTER TO WHERE TO PUT IT
001460  012022                          MOV (R0)+,(R2)+
001462  032705  000002                  BIT     #..ISRC,R5      ;INTEGER SOURCE?
001466  001013                          BNE     3$              ;YES
001470  012022                          MOV (R0)+,(R2)+
001472  032737  000200  000056'        BIT     #FS.DBL,FSTAT   ;DOUBLE PRECISION?
001500  001403                          BEQ     1$              ;NOPE
001502  012022                          MOV (R0)+,(R2)+
001504  012022                          MOV (R0)+,(R2)+
001506  000410                          BR 2$
```

```
                                                                                
001510  005022                  1$:     CLR     (R2)+           ;CLEAR ANY EXTRA GARBAGE
001512  005022                          CLR (R2)+
001514  000405                          BR 2$
                                                                                
001516  032737  000100  000056' 3$:     BIT     #FS.LNG,FSTAT   ;JUMBO SIZED INTEGERS?
001524  001401                          BEQ     2$              ;NO
001526  012022                          MOV     (R0)+,(R2)+     ;YES, COPY ONE MORE WORD
                                                                                
001530                          2$:     RETURN                  ;BACK TO CALING ROUTINE (IE. FSAVIT)
                                                                                
                                                                                
                                                                                
                                ;
                                ;SUBROUTINE TO SET FLAGS IN FFLAGS ACCORDING TO ARGUMENT
                                ;PASSED IN R0.
                                ;
001532                          FSETFL: PUSH    R1              ;SAVE R1
001534  005001                          CLR     R1              ;ASSUME NO FLAGS SET
001536  005700                          TST     R0              ;IS R0 ZERO
001540  001003                          BNE     1$              ;NO
001542  052701  000004                  BIS     #FS.Z,R1        ;YES, SET THE ZERO BIT
001546  005700                          TST     R0              ;AND TEST AGAIN
                                                                                
001550  100002                  1$:     BPL     2$              ;NEGATIVE?
001552  052701  000010                  BIS     #FS.N,R1        ;YES!
001556  010137  000124'          2$:     MOV     R1,FFLAGS       ;SAVE EXIT FLAGS
001562                                  POP     R1              ;RESTORE R1
001564                                  RETURN                  ;AND RETURN
                                                                                
```

```
                                        $CFCC::  PUSH    R0                ;SAVE RETURN POINTER
001570  016600  000002                          MOV     2(SP),R0          ;RESTORE R0
001574  013766  000056'  000002                 MOV     FSTAT,2(SP)       ;COPY FLAGS
001602  042766  177760  000002                 BIC     #177760,2(SP)     ;AND ONLY FLAGS
001610  000002                                  RTI                       ;AND DO RETURN FROM INTERUPT

001612  012737  000004  000126'  $SETF::  MOV     #4,FSIZE          ;4 BYTE FPP #
001620  042737  000200  000056'           BIC     #FS.DBL,FSTAT     ;CLEAR D FLAG
001626  000200                           RTS R0

001630  012737  000010  000126'  $SETD::  MOV     #10,FSIZE         ;8 BYTE FPP #
001636  052737  000200  000056'           BIS     #FS.DBL,FSTAT     ;SET D FLAG
001644  000200                           RTS R0

001646  012737  000002  000130'  $SETI::  MOV     #2,FISIZE         ;2 BYTE INT #
001654  042737  000100  000056'           BIC     #FS.LNG,FSTAT     ;CLEAR L FLAG
001662  000200                           RTS R0

001664  012737  000004  000130'  $SETL::  MOV     #4,FISIZE         ;4 BYTE INT #
001672  052737  000100  000056'           BIS     #FS.LNG,FSTAT     ;SET L FLAG
001700  000200                           RTS R0

001702                                  $LDFPS::
001702                                          CALL    FSAVIT            ;SAVE STUFF
001706  000400                                    ..DST                   ;   GET SRC PTR (LOOKS LIKE DST)
001710  017700  000052'                  MOV     @FDST,R0          ;GET FPP BITS
001714  010037  000056'                  MOV     R0,FSTAT          ;NEW STAT BITS
001720  032700  000200                  BIT     #FS.DBL,R0        ;DOUBLE?
001724  001003                          BNE     1$                ;YES
001726                                  SETF                      ;FLOATING
001732  000402                          BR 2$

001734                                  1$:     SETD                      ;DOUBLE

001740  032700  000100          2$:     BIT     #FS.LNG,R0        ;LONG?
001744  001003                          BNE     3$                ;YES
001746                                  SETI                      ;SHORT
001752  000402                          BR 4$

001754                                  3$:     SETL                      ;LONG

001760  000137  004024'          4$:     JMP     FDONE
```

```
                                
                        $STFPS::
002764                          CALL    FSAVIT          ;SAVE STUFF
002764                                  ..DST           ;   GET DST PTR
002770  000400                  MOV     FSTAT,@FDST     ;SAVE BITS
002772  013777  000056' 000052' JMP     FDONE
002000  000137  004024'
                        $CLRD::
002004                  $CLRF:: CALL    FSAVIT          ;SAVE STUFF
002004                                  ..DST!..FDST! ..PDST!..CFLG
002010  112400                  MOV     #FVALB,R0       ;POINT TO VALUE B
002012  012700  000032'         CLR     (R0)+           ;CLEAR IT OUT
002016  005020                  CLR (R0)+
002020  005020                  CLR (R0)+
002022  005020                  CLR (R0)+
002024  005020                  JMP     FDONE
002026  000137  004024'
                        $TSTF::
002032                  $TSTD:: CALL    FSAVIT          ;SAVE STUFF
002032                                  ..DST!..FDST!..GDST!..CFLG
002036  106400                  JMP     FDONE
002040  000137  004024'
                        $ABSF::
002044                  $ABSD:: CALL    FSAVIT          ;SAVE STUFF
002044                                  ..DST!..FDST!..GDST!..CFLG
002050  106400                  BIC     #100000,@FDST   ;ABS THE #
002052  042777  100000  000052' BIC #100000,FVALB
002060  042737  100000  000032' JMP     FDONE
002066  000137  004024'
                        $NEGF::
002072                  $NEGD:: CALL    FSAVIT          ;SAVE STUFF
002072                                  ..DST!..FDST!..GDST!..CFLG
002076  106400                  MOV     FVALB,R0        ;GET EXPONENT
002100  013700  000032'         BEQ     1$              ;CAN'T NEGATE ZERO!!!!!
002104  001406                  ADD     #100000,R0      ;NEGATE NUMBER
002106  062700  100000          MOV     R0,@FDST        ;PUT BACK NEGATIVE NUMBER
002112  010077  000052'         MOV     R0,FVALB        ;AND SAVE FOR FLAG CHECKING
002116  010037  000032'  1$:     JMP     FDONE
002122  000137  004024'
                        $LDF::
002126                  $LDD::
002126                  $STF::
002126                  $STD::  CALL    FSAVIT
002126                                  ..SRC!..FSRC!..GSRC! ..DST!..FDST! ..PDST!..CFLG
002132  112415          $LDF1:  MOV     #FVALA,R0       ;POINT TO SRC
002134  012700  000016'         MOV     #FVALB,R1       ;POINT TO DST
002140  012701  000032'         MOV     (R0)+,(R1)+     ;COPY TO DST
002144  012021                  MOV     (R0)+,(R1)+     ;COPY TO DST
002146  012021                  MOV     (R0)+,(R1)+     ;COPY TO DST
002150  012021                  MOV     (R0)+,(R1)+     ;COPY TO DST
002152  012021                  JMP     FDONE
002154  000137  004024'
                        $LDEXP::
002160                          CALL    FSAVIT
002160                                  ..SRC!..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
002164  116411                  CALL    FSETB           ;UNPACK DST #
002166                  MOV     @FSRC,FEXPB     ;GET EXP
002172  017737  000046' 000044'
```

```
                                        CALL    FPACKR          ;PACK BACK TOGETHER
002204  000137  004024'                 JMP     FDONE

002210                          $STEXP::        CALL    FSAVIT
002214  100415                                  ..SRC!..FSRC!..GSRC!..DST!..CFLG
002216                                  PUSH    FVALA           ;PUSH PACKED SOURCE
002222                                  CALL    FSETA           ;UNPACK SRC #
002226  013700  000030'                 MOV     FEXPA,R0        ;GET EXPONENT
002232  010077  000052'                 MOV     R0,@FDST        ;SAVE IT AS RESULT
002236                                  CALL    FSETFL          ;AND SET FLAGS FOR RESULT
002242                                  POP     FVALB           ;GET BITS TO CHECK
002246  000137  004024'                 JMP     FDONE

002252                          $LDCIF::
002252                          $LDCID::
002252                          $LDCLF::
002252                          $LDCLD::
002252                                  CALL    FSAVIT
002256  112413                                  ..SRC!..ISRC!..GSRC!..DST!..FDST!..PDST!..CFLG
002260  013737  000020'  000022'         MOV     FMANA,FMANA+2   ;MOVE LOW ORDER LONG INTEGER BITS
002266  013737  000016'  000020'         MOV     FSGNA,FMANA     ;MOVE ARG UP TO MANTISSA
002274  012737  000001   000016'         MOV     #1,FSGNA        ;USE SIGN OF 1
002302  005037  000024'                 CLR     FMANA+4         ;CLEAR LOW PART
002306  005037  000026'                 CLR FMANA+6
002312  032737  000100   000056'         BIT     #FS.LNG,FSTAT   ;LONG INTEGERS?
002320  001407                          BEQ     1$              ;NO
002322  012737  000030   000030'         MOV #30,FEXPA
002330                                  CALL    $$ANSW          ;CONVERT ARG AND MAKE POSITIVE
002334  000137  004024'                 JMP     FDONE           ;FINISH

002340  005037  000022'          1$:    CLR     FMANA+2         ;CLEAR *3* WORDS OF MANTISSA
002344  012737  000010   000030'         MOV     #10,FEXPA       ;EXP=10
002352                                  CALL    $$ANSW          ;CREATE RESULT, NEGATE ARG IF NEGATIVE
002356  000137  004024'                 JMP     FDONE           ;ALL DONE

002362                          $STCFI::
002362                          $STCFL::
002362                          $STCDI::
002362                          $STCDL::
002362                                  CALL    FSAVIT
002366  121415                                  ..SRC!..FSRC!..GSRC!..DST!..IDST!..JDST!..CFLG
002370                                  CALL    FSETA           ;UNPACK #
002374  012705  000020'                 MOV     #FMANA,R5       ;POINT TO MANTISSA
002400  012500                          MOV     (R5)+,R0        ;GET MANTISSA
002402  012501                          MOV (R5)+,R1
002404  012502                          MOV (R5)+,R2
002406  013703  000030'                 MOV     FEXPA,R3        ;GET EXP

002412  020327  000050          1$:    CMP     R3,#50          ;DONE?
002416  002006                          BGE     2$              ;YES
002420  000241                          CLC                     ;DON'T LET A BIT FALL IN
002422  006000                          ROR     R0              ;ROTATE MANTISSA
002424  006001                          ROR R1
002426  006002                          ROR R2
002430  005203                          INC     R3              ;COUNT A ROTATION
002432  000767                          BR      1$              ;CHECK AGAIN
```

```
                                2$:     BGT     FVSET           ;OVERFLOW!
                                        TST     R0              ;ANY BITS LEFT IN HIGHEST?
                                        BNE     FVSET           ;YES, OVERFLOW
                                        BIT     #FS.LNG,FSTAT   ;LONG INTEGERS?
                                        BEQ     3$              ;NO!
                                        TST     FSGNA           ;POSITIVE OR NEGATIVE
                                        BGE     4$              ;POSITIVE!
                                        NEG     R2              ;NEGATE NUMBER
                                        ADC     R1
                                        BCS 4$
                                        NEG R2
                                4$:     MOV     R1,FVALB        ;SAVE AS RESULT
                                        MOV R2,FVALB+2
                                        MOV     R2,R0           ;COPY LSB
                                        SWAB    R0              ;PUT TWO BYTES TOGETHER
                                        BIS R2,R0
                                        BIC     #UPPER,R0       ;SAVE ONLY UPPER BYTE
                                        BIS     R1,R0           ;PUT MSB WITH IT
                                        CALL    FSETFL          ;SET FLAGS
                                        JMP     FDONE

                                3$:     TST     R1              ;ONE WORD?
                                        BNE     FVSET           ;NO, OVERFLOW!
                                        TST     FSGNA           ;SIGN POSITIVE?
                                        BGE     5$              ;YES
                                        NEG     R2              ;NEGATE NUMBER
                                5$:     MOV     R2,FVALB        ;SAVE AS RESULT
                                        MOV     R2,R0           ;COPY TO SET FLAGS
                                        CALL    FSETFL          ;SET FLAGS
                                        JMP     FDONE

                        FVSET:  BIS     #FS.V,FSTAT     ;SET VV FLAG
                                BIS     #FS.V,FFLAGS    ;AND RETURNED FLAGS
                                JMP     FDONE1
```

```
002576                          $LDCFD::
002576                          $LDCDF::
002576                                  PUSH    R1                      ;SAVE A REG
002600  013701  000056'                MOV     FSTAT,R1                ;GET STATUS BITS
002604  010137  000060'                MOV     R1,FSTATS               ;SAVE STATUS
002610  005101                        COM     R1                      ;COMPLEMENT STATUS BITS
002612  042701  177577                BIC     #^CFS.DBL,R1            ;LEAVE ONLY DOUBLE BIT
002616  042737  000200  000056'       BIC     #FS.DBL,FSTAT           ;CLEAR IT OUT OF STATUS
002624  050137  000056'                BIS     R1,FSTAT                ;SET/CLEAR BIT IN STATUS (COMP OF BEFORE)
002630                                  POP     R1                      ;RESTORE REG
002632                                  CALL    FSAVIT                  ;SAVE STUFF
002636  110015                              ..SRC!..FSRC!..GSRC!..PDST!..CFLG
002640  012705  000005                MOV     #..SRC!..FSRC,R5        ;GET FLAG BITS FOR DST
002644  013704  000114'                MOV     REG7,R4                 ;GET PSEUDO PC
002650  013703  000116'                MOV     FMOV,R3                 ;ALSO DST BITS
002654  012702  000052'                MOV     #FDST,R2                ;LAST BUT NOT LEAST, WHERE TO PUT INFO
002660  013737  000060' 000056'       MOV     FSTATS,FSTAT            ;RESTORE PROPER STATUS
002666                                  CALL    FPTR                    ;GET POINTER
002672  000137  002134'                JMP     $LDF1                   ;PUT IN DST
002676
002676                          $STCDF::
002676                          $STCFD::
002676                                  CALL    FSAVIT                  ;SAVE STUFF
002702  100015                              ..SRC!..FSRC!..GSRC!..CFLG
002704  013701  000056'                MOV     FSTAT,R1                ;GET STATUS BITS
002710  010137  000060'                MOV     R1,FSTATS               ;SAVE
002714  010102                        MOV     R1,R2                   ;COPY
002716  005101                        COM     R1                      ;COM ALL BITS
002720  042701  177577                BIC     #^CFS.DBL,R1            ;LEAVE DOUBLE BIT ONLY
002724  042702  000200                BIC     #FS.DBL,R2              ;CLEAR OUT OF COPY
002730  050102                        BIS     R1,R2                   ;COMP BIT IN COPY
002732  010137  000056'                MOV     R1,FSTAT                ;SAVE NEW STATUS
002736  012705  012400                MOV     #..DST!..FDST!..PDST,R5;BITS TO GET DST POINTER
002742  013704  000114'                MOV     REG7,R4                 ;GET PSEUDO PC
002746  013703  000116'                MOV     FMOV,R3                 ;DST BITS
002752  012702  000052'                MOV     #FDST,R2                ;WHERE TO PUT INFO
002756                                  CALL    FPTR                    ;GET POINTER
002762  012700  000016'                MOV     #FVALA,R0               ;GET POINTER TO VALUE
002766  013701  000052'                MOV     FDST,R1                 ;POINTER TO WHERE TO PUT STUFF
002772  011021                        MOV     (R0),(R1)+              ;COPY TO RESULT
002774  012037  000032'                MOV     (R0)+,FVALB             ;COPY FOR FLAGS SETTING
003000  012021                        MOV     (R0)+,(R1)+             ;COPY TO RESULT
003002  032737  000200  000056'       BIT     #FS.DBL,FSTAT           ;SAVING DOUBLE?
003010  001402                        BEQ     1$                      ;NO
003012  012021                        MOV     (R0)+,(R1)+             ;COPY REST
003014  012021                        MOV     (R0)+,(R1)+
003016  013737  000060' 000056' 1$:   MOV     FSTATS,FSTAT            ;RESTORE CORRECT STATUS
003024  000137  004024'                JMP     FDONE                   ;FINIS!
```

```
                                                
003030                          $CMPF::
003030                          $CMPD::
003030                                  CALL    FSAVIT
003034  106415                              ..SRC!..FSRC!..GSRC!..DST!..FDST!..GDST!..CFLG
003036  062737  100000  000032'          ADD     #100000,FVALB   ;NEGATE SECOND #
003044                                  CALL    $$ADD
003050  000137  004024'                  JMP     FDONE

003054                          $SUBF::
003054                          $SUBD::
003054                                  CALL    FSAVIT
003060  116415                              ..SRC!..FSRC!..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
003062  062737  100000  000016'          ADD #100000,FVALA
003070                                  CALL    $$ADD
003074  000137  004024'                  JMP     FDONE

003100                          $ADDF::
003100                          $ADDD::
003100                                  CALL    FSAVIT
003104  116415                              ..SRC!..FSRC:..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
003106                                  CALL    $$ADD
003112  000137  004024'                  JMP     FDONE

003116                          $MULF::
003116                          $MULD::
003116                                  CALL    FSAVIT
003122  116415                              ..SRC!..FSRC!..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
003124                                  CALL    $$MUL
003130                                  CALL    FNORMB
003134                                  CALL    FPACKB
003140  000137  004024'                  JMP     FDONE

003144                          $MODF::
003144                          $MODD::
003144                                  CALL    FSAVIT
003150  116415                              ..SRC!..FSRC!..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
003152                                  CALL    $$MUL
003156                                  CALL    FNORM           ;NORMALIZE RESULT
003162  012700  000006                  MOV #6,R0
003166  012701  000016'                 MOV #FVALA,R1
003172  012702  000032'                 MOV #FVALB,R2
003176  012221                  1$:     MOV     (R2)+,(R1)+     ;COPY RESULT
003200                                  SOB R0,1$

003204  005002                          CLR     R2              ;CLEAR OFFSET INTO MANTISSA
003206  005000                          CLR     R0              ;CLEAR COUNT OF EXP
003210  012701  000200                  MOV     #200,R1         ;BIT TO CLEAR
003214  013703  000044'                 MOV     FEXPB,R3        ;GET B'S EXPONENT

003220                          2$:
003220  020003                          CMP     R0,R3           ;INTO FRACTIONAL PART OF #?
003222  002003                          BGE     3$              ;YES
003224  040162  000034'                 BIC     R1,FMANB(R2)    ;CLEAR PART OF INTEGER
003230  000402                          BR      4$              ;NEXT BIT!
003232  040162  000020'          3$:     BIC     R1,FMANA(R2)    ;CLEAR PART OF FFRACTION

```

```
003236  005200                  4$:     INC     R0              ;COUNT A SHIFT
003240  000241                          CLC                     ;DON'T WANT EXTRA BITS
003242  006001                          ROR     R1              ;SHIFT BIT
003244  001365                          BNE     2$              ;RESET TO FIRST BIT?
003246  006001                          ROR     R1              ;GET BACK HIGH ORDER BIT
003250  062702  000002                  ADD     #2,R2           ;POINT TO NEXT WORD OF MANTISSA
003254  020227  000010                  CMP     R2,#10          ;END OF MANTISSA?
003260  002757                          BLT     2$              ;NOPE, DO THIS WORD
003262  023727  000052' 000132'         CMP     FDST,#FAC0      ;EVEN AC?
003270  001404                          BEQ     5$              ;YES!
003272  023727  000052' 000152'         CMP     FDST,#FAC2      ;CHECK AGAIN, IS IT?
003300  001016                          BNE     6$              ;NO!

003302                          5$:     CALL    FNORMA          ;NORMALIZE A
003306                                  CALL    FPACKA          ;PACK UP THE #
003312  012700  000016'                 MOV     #FVALA,R0       ;POINT TO INT PART
003316  013701  000052'                 MOV     FDST,R1         ;POINT TO DESTINATION AC
003322  062701  000010                  ADD     #10,R1          ;POINT TO ONE AFTER IT
003326  012021                          MOV     (R0)+,(R1)+     ;COPY OUT #
003330  012021                          MOV (R0)+,(R1)+
003332  012021                          MOV (R0)+,(R1)+
003334  012021                          MOV (R0)+,(R1)+

003336                          6$:     CALL    FNORMB          ;NORMALIZE B
003342                                  CALL    FPACKB          ;PACK THE FRACTION
003346  000137  004024'                 JMP     FDONE           ;SAVE THE FRACTION

003352                          $DIVF::
003352                          $DIVD::
003352                                  CALL    FSAVIT
003356  116415                              ..SRC!..FSRC!..GSRC!..DST!..FDST!..GDST!..PDST!..CFLG
003360                                  CALL    $$DIV
003364  000137  004024'                 JMP     FDONE
```

```
                                        ;
                                        ;UNPACK FLOATING POINT NUMBER
                                        ;
003370  012700  000016'       FSETA:  MOV #FVALA,R0
003374  000402                        BR FSET

003376  012700  000032'       FSETB:  MOV #FVALB,R0

003402  010001                FSET:   MOV R0,R1
003404  012702  000062'               MOV #FTMP,R2
003410  010203                        MOV R2,R3
003412  012123                        MOV (R1)+,(R3)+
003414  012123                        MOV (R1)+,(R3)+
003416  012123                        MOV (R1)+,(R3)+
003420  012123                        MOV (R1)+,(R3)+
003422  005010                        CLR (R0)
003424  012201                        MOV (R2)+,R1
003426  010105                        MOV R1,R5
003430  001411                        BEQ 3$
003432  002402                        BLT 1$
003434  005220                        INC (R0)+
003436  000401                        BR 2$

003440  005320                1$:     DEC (R0)+

003442  042701  177600        2$:     BIC #^C177,R1
003446  052701  000200                BIS #200,R1
003452  000401                        BR 4$

003454  005020                3$:     CLR (R0)+

003456  010120                4$:     MOV R1,(R0)+
003460  012220                        MOV (R2)+,(R0)+
003462  032737  000200  000056'       BIT #FS.DBL,FSTAT
003470  001403                        BEQ 5$
003472  012220                        MOV (R2)+,(R0)+
003474  012220                        MOV (R2)+,(R0)+
003476  000402                        BR 6$

003500  005020                5$:     CLR (R0)+
003502  005020                        CLR (R0)+

003504  010501                6$:     MOV R5,R1
003506  006301                        ASL R1
003510  105001                        CLRB R1
003512  000301                        SWAB R1
003514  162701  000200                SUB #200,R1
003520  010120                        MOV R1,(R0)+
003522                                RETURN
```

```
                                        ;
                                        ;NORMALIZE FLOATING POINT NUMBER
                                        ;
003524  012700  000020'        FNORMA: MOV #FMANA,R0
003530  000402                        BR FNORM

003532  012700  000034'        FNORMB: MOV #FMANB,R0

003536  012001                FNORM:  MOV (R0)+,R1
003540  010105                        MOV R1,R5
003542  012002                        MOV (R0)+,R2
003544  050205                        BIS R2,R5
003546  012003                        MOV (R0)+,R3
003550  050305                        BIS R3,R5
003552  012004                        MOV (R0)+,R4
003554  050405                        BIS R4,R5
003556  001003                        BNE 1$
003560  005060  177766                CLR -12(R0)
003564                                RETURN

003566  032701  177400        1$:     BIT #^C377,R1
003572  001407                        BEQ 2$
003574  000241                        CLC
003576  006001                        ROR R1
003600  006002                        ROR R2
003602  006003                        ROR R3
003604  006004                        ROR R4
003606  005210                        INC (R0)
003610  000766                        BR 1$

003612  032701  000200        2$:     BIT #200,R1
003616  001006                        BNE 3$
003620  006304                        ASL R4
003622  006103                        ROL R3
003624  006102                        ROL R2
003626  006101                        ROL R1
003630  005310                        DEC (R0)
003632  000767                        BR 2$

003634  010440                3$:     MOV R4,-(R0)
003636  010340                        MOV R3,-(R0)
003640  010240                        MOV R2,-(R0)
003642  010140                        MOV R1,-(R0)
003644                                RETURN
```

```
                                        ;
                                        ;PACK A NUMBER TO FLOATING FORMAT AGAIN
                                        ;
003646  012702  000016'       FPACKA: MOV #FVALA,R2
003652  000402                       BR FPACK

003654  012702  000032'       FPACKB: MOV #FVALB,R2

003660  010203                FPACK:  MOV R2,R3
003662  012700  000062'               MOV #FTMP,R0
003666  012701  000006               MOV #6,R1
003672  012320                1$:     MOV (R3)+,(R0)+
003674                               SOB R1,1$

003700  012700  000062'               MOV #FTMP,R0
003704  005710                       TST (R0)
003706  001435                       BEQ FUNDER
003710  013701  000074'               MOV FTMP+12,R1
003714  020127  000177               CMP R1,#177
003720  003035                       BGT FOVER
003722  020127  177601               CMP R1,#-177
003726  002425                       BLT FUNDER
003730  062701  000200               ADD #200,R1
003734  000301                       SWAB R1
003736  000241                       CLC
003740  006001                       ROR R1
003742  005720                       TST (R0)+
003744  002002                       BGE 2$
003746  052701  100000               BIS #100000,R1

003752  042710  177600       2$:     BIC #^C177,(R0)
003756  052001                       BIS (R0)+,R1
003760  010122                       MOV R1,(R2)+
003762  012022                       MOV (R0)+,(R2)+
003764  032737  000200  000056'       BIT #FS.DBL,FSTAT
003772  001402                       BEQ 3$
003774  012022                       MOV (R0)+,(R2)+
003776  012022                       MOV (R0)+,(R2)+

004000                       3$:     RETURN

004002  005022                FUNDER: CLR (R2)+
004004  005022                       CLR (R2)+
004006  005022                       CLR (R2)+
004010  005022                       CLR (R2)+
004012                               RETURN

004014  052737  000002  000056' FOVER:  BIS #FS.V,FSTAT
004022  000767                       BR FUNDER
```

```
                                        ;
                                        ;FINISH UP FPP INSTRUCTION
                                        ;
004024  032737  100000  000120' FDONE:  BIT     #..CFLG,FDESC   ;CHECK FLAGS?
004032  001415                          BEQ     FDONE1          ;NO
004034  013700  000056'                 MOV FSTAT,R0
004040  013701  000032'                 MOV FVALB,R1
004044  001003                          BNE     1$              ;SET Z?
004046  052700  000004                  BIS     #FS.Z,R0        ;YES
004052  005701                          TST R1
004054  100002                  1$:     BPL     2$              ;SET N?
004056  052700  000010                  BIS     #FS.N,R0        ;YES
                                
004062  010037  000056'          2$:     MOV R0,FSTAT
                                
004066  012700  000032'          FDONE1: MOV     #FVALB,R0       ;POINT TO RESULT, IF ANY
004072  013701  000052'                  MOV     FDST,R1         ;AND DST, TOO
004076  032737  010000  000120'          BIT     #..PDST,FDESC   ;SAVE RESULT (FLOATING)?
004104  001411                          BEQ     1$              ;NO
004106  012021                          MOV (R0)+,(R1)+
004110  012021                          MOV (R0)+,(R1)+
004112  032737  000200  000056'          BIT #FS.DBL,FSTAT
004120  001415                          BEQ 2$
004122  012021                          MOV (R0)+,(R1)+
004124  012021                          MOV (R0)+,(R1)+
004126  000412                          BR 2$
                                
004130  032737  020000  000120' 1$:     BIT     #..JDST,FDESC   ;SAVE RESULT (INTEGER)?
004136  001406                          BEQ     2$              ;NO
004140  012021                          MOV (R0)+,(R1)+
004142  032737  000100  000056'          BIT #FS.LNG,FSTAT
004150  001401                          BEQ 2$
004152  012021                          MOV (R0)+,(R1)+
                                
004154  012700  000114'          2$:     MOV     #REG7,R0        ;POINT TO SAVED REGISTERS
004160  014006                          MOV     -(R0),SP        ;RESTORE SP
004162  014005                          MOV     -(R0),R5        ;AND R5
004164  014004                          MOV     -(R0),R4        ;   AND R4
004166  014003                          MOV     -(R0),R3        ;       AND R3
004170  014002                          MOV     -(R0),R2        ;           AND R2
004172  014001                          MOV     -(R0),R1        ;               AND R1
004174  014000                          MOV     -(R0),R0        ;                   AND FINALLY, R0
004176                                  PUSH <FFLAGS,REG7>
004206  000002                          RTI                     ;RETURN
```

```
                                        ;
                                        ;   $$ADD -- ADD FVALA TO FVALB
                                        ;
                                        ;
                                        ;UNPACK NUMBERS AND CHECK IF EITHER IS ZERO.  IF ONE IS, THE ANSWER IS
                                        ;JUST THE OTHER NUMBER.  (ADDITIVE IDENTITY.)
                                        ;
                                                .ENABL LSB

004210                                  $$ADD:  CALL    FSETA           ;UNPACK A
004214                                          CALL    FSETB           ;UNPACK B
004220  005737  000016'                         TST     FSGNA           ;IS A=0?
004224  001003                                  BNE     2$              ;NOPE

004226                                  1$:     CALL    FPACKB          ;REPACK B
004232                                          RETURN                  ;AND RETURN IT AS THE ANSWER

004234  005737  000032'                 2$:     TST     FSGNB           ;IS B=0?
004240  001013                                  BNE     4$              ;NOPE

004242                                  3$:     CALL    FPACKA          ;REPACK A
004246  012701  000016'                         MOV     #FVALA,R1       ;POINT TO A
004252  012702  000032'                         MOV     #FVALB,R2       ;POINT TO B
004256  012122                                  MOV     (R1)+,(R2)+     ;MOVE A INTO B
004260  012122                                  MOV (R1)+,(R2)+
004262  012122                                  MOV (R1)+,(R2)+
004264  012122                                  MOV (R1)+,(R2)+
004266                                          RETURN                  ;AND RETURN B AS ANSWER
                                        ;
                                        ;COMPARE EXPONENTS TO CHECK WHICH VALUE MUST BE SHIFTED
                                        ;RIGHT TO ALLIGN THE TWO NUMBERS SO THAT THEY MAY BE ADDED.
                                        ;
004270  013700  000030'                 4$:     MOV     FEXPA,R0        ;GET EXPONENT FOR A
004274  163700  000044'                         SUB     FEXPB,R0        ;MINUS B'S EXPONENT
004300  002407                                  BLT     5$              ;B HAS LARGER EXPONENT
004302  001470                                  BEQ     $$ADD1          ;NO SHIFTING NEEDED!
004304  020027  000070                         CMP     R0,#56.         ;NUMBER OUT OF RANGE?
004310  002354                                  BGE     3$              ;YES, SO JUST RETURN VALUE OF B
004312  012705  000034'                         MOV     #FMANB,R5       ;NO, SHIFT B RIGHT TO ALLIGN NUMBERS
004316  000406                                  BR 6$

004320  005400                          5$:     NEG     R0              ;A HAS LARGER EXPONENT
004322  020027  000070                         CMP     R0,#56.         ;OUT OF RANGE?
004326  002337                                  BGE     1$              ;YES, SO JUST RETURN VALUE OF B
004330  012705  000020'                         MOV     #FMANA,R5       ;NO, SHIFT A RIGHT TO ALLIGN NUMBERS
                                        ;
                                        ;PICK UP THE NUMBER TO SHIFT
                                        ;
004334  012501                          6$:     MOV (R5)+,R1
004336  012502                                  MOV (R5)+,R2
004340  012503                                  MOV (R5)+,R3
004342  012504                                  MOV (R5)+,R4
004344  060015                                  ADD     R0,@R5          ;RESULTING EXPONENT AFTER SHIFT
```

```
1147                                    ;SHIFTING IN MULTIPLES OF 16 PLACES SIMPLY BUMPS THE
1148                                    ;REGISTERS.
1149                                    ;
1150 004346  020027  000020            7$:     CMP     R0,#16.         ;SHIFTING AT LEAST 16 PLACES?
1151 004352  002410                            BLT     8$              ;NOPE
1152 004354  010304                            MOV     R3,R4           ;YES, SO DO VERY FASH SHIFT
1153 004356  010203                            MOV R2,R3
1154 004360  010102                            MOV R1,R2
1155 004362  005001                            CLR     R1              ;ZEROES INTO R1
1156 004364  162700  000020                    SUB     #16.,R0         ;COUNT 16 SHIFTS
1157 004370  003366                            BGT     7$              ;MORE TO SHIFT
1158 004372  001430                            BEQ     10$             ;NUMBERS ARE ALLIGNED!
1159                                    ;
1160                                    ;IF SHIFTING AT LEAST 8 PLACES, WE CAN USE A CLASSY
1161                                    ;BYTE ALGORITHM.
1162                                    ;
1163 004374  020027  000010            8$:     CMP     R0,#8.          ;SHIFTING AT LEAST 8 PLACES?
1164 004400  002416                            BLT     9$              ;NOPE
1165 004402  105004                            CLRB    R4              ;CLEAR RIGHT BYTE OF R4
1166 004404  150304                            BISB    R3,R4           ;MOVE LOWER BYTE OF R3 INTO HOLE
1167 004406  000304                            SWAB    R4              ;REVERSE BYTES (SHIFT RIGHT 1 BYTE!!!)
1168 004410  105003                            CLRB    R3              ;ETC.
1169 004412  150203                            BISB R2,R3
1170 004414  000303                            SWAB R3
1171 004416  105002                            CLRB R2
1172 004420  150102                            BISB R1,R2
1173 004422  000302                            SWAB R2
1174 004424  105001                            CLRB    R1              ;ZEROES FILL R1
1175 004426  000301                            SWAB R1
1176 004430  162700  000010                    SUB     #8.,R0          ;COUNT THE SHIFT
1177 004434  001407                            BEQ     10$             ;ALL DONE
1178                                    ;
1179                                    ;SHIFTING HAS NOW BEEN REDUCED TO CASE OF 7 PLACES OR LESS.
1180                                    ;USE THE GRUNGY SLOW METHOD...
1181                                    ;
1182 004436  000241                    9$:     CLC                     ;ZEROES FILL R1
1183 004440  006001                            ROR     R1              ;BUMP EVERYTHING RIGHT 1 BIT
1184 004442  006002                            ROR R2
1185 004444  006003                            ROR R3
1186 004446  006004                            ROR     R4              ;WITH BITS FALLING OUT OF R4
1187 004450                                    SOB     R0,9$           ;7 SHIFTS MAX
1188                                    ;
1189                                    ;PUT SHIFTED ARGUMENT BACK WHERE IT CAME FROM
1190                                    ;
1191 004454  010445                    10$:    MOV R4,-(R5)
1192 004456  010345                            MOV R3,-(R5)
1193 004460  010245                            MOV R2,-(R5)
1194 004462  010145                            MOV R1,-(R5)
1195                                    ;
1196                                    ;PERFORM THE ADDITION
1197                                    ;
1198 004464  012701  000030'           $$ADD1: MOV     #FEXPA,R1       ;POINT PAST A
1199 004470  012702  000044'                   MOV     #FEXPB,R2       ;AND PAST B
1200 004474  023737  000016' 000032'           CMP     FSGNA,FSGNB     ;ADDING OR SUBTRACTING?
1201 004502  001016                            BNE     14$             ;SUBTRACTING
1202 004504  000241                            CLC                     ;ADDING NUMBERS OF SAME SIGN
```

```
1204                                    ;DO MULTIPLE PRECISION BINARY ADD...
1205                                    ;
1206 004506  064241            11$:     ADD     -(R2),-(R1)     ;ADD LOW ORDER WORDS
1207 004510  010103                     MOV     R1,R3           ;COPY POINTER
1208 004512  005543                     ADC     -(R3)           ;PROPAGATE CARRY
1209 004514  005543                     ADC     -(R3)
1210 004516  005543                     ADC     -(R3)
1211 004520  064241                     ADD     -(R2),-(R1)     ;ADD NEXT TO LAST WORDS
1212 004522  010103                     MOV     R1,R3           ;COPY POINTER
1213 004524  005543                     ADC     -(R3)           ;PROPAGATE CARRY
1214 004526  005543                     ADC     -(R3)
1215 004530  064241                     ADD     -(R2),-(R1)     ;ADD SECOND WORDS
1216 004532  005541                     ADC     -(R1)           ;PLUS CARRY
1217 004534  064211                     ADD     -(R2),@R1       ;AND FINAL WORD
1218 004536  000417                     BR      18$             ;AND FINISH
1219                                    ;
1220                                    ;DO MULTIPLE PRECISION BINARY SUBTRACTION...
1221                                    ;
1222 004540  164241            14$:     SUB     -(R2),-(R1)     ;SUBTRACT LOW ORDER WORDS
1223 004542  010103                     MOV     R1,R3           ;COPY POINTER
1224 004544  005643                     SBC     -(R3)           ;DO BORROWING
1225 004546  005643                     SBC  -(R3)
1226 004550  005643                     SBC  -(R3)
1227 004552  164241                     SUB     -(R2),-(R1)     ;SUBTRACT SECOND FROM LAST WORDS
1228 004554  010103                     MOV     R1,R3           ;COPY POINTER
1229 004556  005643                     SBC     -(R3)           ;BORROW
1230 004560  005643                     SBC  -(R3)
1231 004562  164241                     SUB     -(R2),-(R1)     ;SUBTRACT SECOND WORDS
1232 004564  005641                     SBC     -(R1)           ;BORROW
1233 004566  164211                     SUB     -(R2),@R1       ;FINAL SUBTRACTION
1234 004570  000402                     BR      18$             ;AND FINISH
1235                                    ;
1236                                    ;DONE PROCESSING.  NEGATE RESULT IF NECESSARY.
1237                                    ;
1238 004572  012701  000020'   $$ANSW: MOV     #FMANA,R1       ;POINT TO RESULT
1239
1240 004576  005711            18$:     TST     @R1             ;END UP NEGATIVE?
1241 004600  002013                     BGE     $$ADD4          ;NO
1242 004602  012701  000030'            MOV     #FEXPA,R1       ;YES, POINT PAST NUMBER
1243 004606  012700  000004             MOV     #4,R0           ;WHICH IS FOUR WORDS LONG
1244 004612  000241                     CLC                     ;NO CARRY
1245
1246 004614  005541            $$ADD2:  ADC     -(R1)           ;ADD CARRY
1247 004616  103401                     BCS     $$ADD3          ;GENERATE CARRY?
1248 004620  005411                     NEG     @R1             ;NO, NEGATE THAT WORD
1249 004622                    $$ADD3:  SOB     R0,$$ADD2       ;LOOP
1250 004626  005441                     NEG     -(R1)           ;NEGATE SIGN
1251
1252 004630                    $$ADD4:  CALL    FNORMA          ;NORMALIZE THE RESULT
1253 004634  000137  004242'            JMP     3$              ;AND RETURN A AS THE RESULT
1254                                    .DSABL LSB
```

```
                                        ;   $$MUL -- MULTIPLY FVALA BY FVALB.  RESULT GOES IN FVALB.
                                        ;
                                        ;
                                        ;FIRST UNPACK ARGUMENTS AND SET UP COUNT FOR WORD LENGTH.
                                        ;
004640                                  $$MUL:  CALL    FSETA           ;UNPACK A
004644                                          CALL    FSETB           ;UNPACK B
004650                                          CALL    FSGNX           ;CALCULATE SIGN OF RESULT
004654  012705  000044'                         MOV     #FEXPB,R5       ;POINT TO ARGUMENT TO TEST
004660  063737  000030'  000044'               ADD     FEXPA,FEXPB     ;MULTIPLICATION ADDS THE EXPONENTS
004666  005337  000044'                         DEC     FEXPB           ;MINUS FUDGE FACTOR
004672  012737  000070  000062'               MOV     #56.,FTMP       ;NUMBER OF BITS
004700  032737  000200  000056'               BIT     #FS.DBL,FSTAT   ;DOUBLE PRECISION?
004706  001005                                  BNE     1$              ;YES
004710  162705  000004                         SUB     #4,R5           ;NO, BACK UP THE POINTER BY 2 WORDS
004714  012737  000030  000062'               MOV     #24.,FTMP       ;FEWER BITS
                                        ;
                                        ;CLEAR RESULT
                                        ;
004722  005000                          1$:     CLR R0
004724  005001                                  CLR R1
004726  005002                                  CLR R2
004730  005003                                  CLR R3
004732  005004                                  CLR R4
                                        ;
                                        ;GENERATE NEXT BIT TO CHECK.  IF BIT OVERFLOWS WORD, THEN CHECK
                                        ;NEXT WORD IN B.  NOTE:  A ZERO WORD IT OPTIMIZED HERE.
                                        ;
004734  006300                          2$:     ASL     R0              ;BUMP BIT LEFT
004736  001036                                  BNE     5$              ;IF THERE STILL IS ONE
004740  005200                                  INC     R0              ;BIT RETURNS TO RIGHT
004742  005745                          3$:     TST     -(R5)           ;GET NEW WORD TO CHECK
004744  001033                                  BNE     5$              ;IT IS NOT ZERO
004746  022737  000020  000062'               CMP     #16.,FTMP       ;LAST WORD?
004754  003027                                  BGT     5$              ;YES, BE CAREFUL
004756  010304                                  MOV     R3,R4           ;NO, SO BUMP WORDS DOWN
004760  010203                                  MOV R2,R3
004762  010102                                  MOV R1,R2
004764  005001                                  CLR     R1              ;ZEROES INTO R1
004766  162737  000020  000062'               SUB     #16.,FTMP       ;COUNT BITS SHIFTED
004774  001362                                  BNE     3$              ;MORE TO DO
                                        ;
                                        ;PUT RESULT INTO FVALB (UNNORMALIZED!)
                                        ;
005000  032704  000001                  4$:     BIT     #1,R4           ;LAST BIT SET?
005002  001405                                  BEQ     10$             ;NO
005004  062704  000001                         ADD     #1,R4           ;YES, ROUND UP
005010  005503                                  ADC     R3              ;WITH CARRY
005012  005502                                  ADC     R2
005014  005501                                  ADC     R1
005016  012700  000034'                  10$:    MOV     #FMANB,R0       ;POINT TO RESULT
005022  010120                                  MOV     R1,(R0)+        ;STORE RESULT
005024  010220                                  MOV R2,(R0)+
005026  010320                                  MOV R3,(R0)+
005030  010420                                  MOV R4,(R0)+
005032                                          RETURN                  ;AND RETURN
```

```
1313                                    ;
1314                                    ;ROTATE RESULT 1 RIGHT AND ADD IF CORRESPONDING BIT SET IN FVALB.
1315                                    ;
1316 005034  000241                    5$:     CLC                     ;CLEAR CARRY
1317 005036  006001                            ROR     R1              ;BUMP EVERYTHING RIGHT
1318 005040  006002                            ROR R2
1319 005042  006003                            ROR R3
1320 005044  006004                            ROR R4
1321 005046  030015                            BIT     R0,@R5          ;NEED TO ADD?
1322 005050  001004                            BNE     7$              ;YES
1323 005052  005337  000062'           6$:     DEC     FTMP            ;COUNT THIS BIT
1324 005056  003326                            BGT     2$              ;AND CONTINUE
1325 005060  000746                            BR      4$              ;ELSE DONE
1326                                    ;
1327                                    ;ADD A TO RESULT
1328                                    ;
1329 005062                            7$:     PUSH    R0              ;SAVE BIT MASK
1330 005064  012700  000030'                   MOV     #FEXPA,R0       ;POINT PAST A
1331 005070  064004                            ADD     -(R0),R4        ;ADD LOW ORDER BITS
1332 005072  005503                            ADC     R3              ;PROPAGATE CARRY
1333 005074  005502                            ADC     R2
1334 005076  005501                            ADC     R1
1335 005100  064003                            ADD     -(R0),R3        ;NEXT TO LAST WORDS
1336 005102  005502                            ADC     R2              ;WITH CARRY
1337 005104  005501                            ADC     R1
1338 005106  064002                            ADD     -(R0),R2        ;SECOND WORDS
1339 005110  005501                            ADC     R1              ;AND CARRY
1340 005112  064001                            ADD     -(R0),R1        ;FIRST WORD
1341 005114                                    POP     R0              ;RESTORE MASK
1342 005116  000755                            BR      6$              ;AND COUNT THE BIT
1343
1344
1345
1346                                    ;
1347                                    ;SUBROUTINE TO CALCULATE SIGN OF RESULT OF MULTIPLY OR DIVIDE
1348                                    ;
1349 005120  023737  000016' 000032'   FSGNX:  CMP     FSGNA,FSGNB     ;SIGNS THE SAME?
1350 005126  001404                            BEQ     1$              ;YES, SO RESULT IS POSITIVE
1351 005130  012737  177777 000032'            MOV     #-1,FSGNB       ;NO, RESULT IS NEGATIVE
1352 005136  000403                            BR 2$
1353
1354 005140  012737  000001 000032'   1$:     MOV     #1,FSGNR        ;RESULT IS POSITIVE
1355 005146                            2$:     RETURN
```

```
                                        ;
                                        ;   $$DIV -- DIVIDE FVALB BY FVALA.  RESULT IN FVALB
                                        ;
                                        ;FIRST, CHECK SIGN OF RESULT AND PICK UP FVALB
                                        ;
005150                                  $$DIV:  CALL    FSETA           ;UNPACK A
005154                                          CALL    FSETB           ;UNPACK B
005160  005737  000016'                         TST     FSGNA           ;DIVISION BY ZERO?
005164  001002                                  BNE     1$              ;NO
005166  000137  004014'                         JMP     FOVER           ;YES! OVERFLOW
                                        ;
005172  163737  000030' 000044'        1$:     SUB     FEXPA,FEXPB     ;DIVISION SUBTRACTS THE EXPONENTS
005200                                          CALL    FSGNX           ;CALCULATE SIGN OF RESULT
005204  012700  000034'                         MOV     #FMANB,R0       ;POINT TO B
005210  011001                                  MOV     @R0,R1          ;PICK UP HIGH ORDER WORD
005212  005020                                  CLR     (R0)+           ;MAKE ROOM FOR RESULT
005214  011002                                  MOV     @R0,R2          ;...
005216  005020                                  CLR (R0)+
005220  011003                                  MOV @R0,R3
005222  005020                                  CLR (R0)+
005224  011004                                  MOV @R0,R4
005226  005010                                  CLR @R0
005230  012705  000034'                         MOV     #FMANB,R5       ;POINT TO RESULT
005234  012746  000400                         MOV     #400,-(SP)      ;BIT MASK
                                        ;
                                        ;CHECK NUMBERS FOR POSSIBILITY OF SUBTRACTION.
                                        ;
005240  012700  000020'                 10$:    MOV     #FMANA,R0       ;POINT TO A
005244  022001                                  CMP     (R0)+,R1        ;CHECK HIGH ORDER WORD
005246  002411                                  BLT     2$              ;BACON, LETTUCE, & TOMATOE
005250  003025                                  BGT     3$              ;...
005252  022002                                  CMP (R0)+,R2
005254  103406                                  BLO 2$
005256  101022                                  BHI 3$
005260  022003                                  CMP (R0)+,R3
005262  103403                                  BLO 2$
005264  101017                                  BHI 3$
005266  022004                                  CMP (R0)+,R4
005270  101015                                  BHI 3$
                                        ;
                                        ;SUBTRACT A FROM B
                                        ;
005272  012700  000030'                 2$:     MOV     #FEXPA,R0       ;POINT PAST A
005276  164004                                  SUB     -(R0),R4        ;SUBTRACT LOW ORDER WORDS
005300  005603                                  SBC     R3              ;WITH BORROW
005302  005602                                  SBC R2
005304  005601                                  SBC R1
005306  164003                                  SUB     -(R0),R3        ;THIRD WORDS
005310  005602                                  SBC R2
005312  005601                                  SBC R1
005314  164002                                  SUB     -(R0),R2        ;SECOND WORDS
005316  005601                                  SBC R1
005320  164001                                  SUB     -(R0),R1        ;FIRST WORD
005322  051615                                  BIS     @SP,@R5         ;MARK SUBTRACTION BIT IN RESULT
```

```
1413                                    ;
1414                                    ;ROTATE A LEFT 1 BIT
1415                                    ;
1416 005324  006304                     3$:     ASL R4
1417 005326  006103                             ROL R3
1418 005330  006102                             ROL R2
1419 005332  006101                             ROL R1
1420 005334  000241                             CLC                     ;WIPE OUT THE CARRY
1421 005336  006016                             ROR     @SP             ;ROTATE MASK 1 RIGHT
1422 005340  001337                             BNE     10$             ;KEEP ON TRUCKIN
1423 005342  006016                             ROR     @SP             ;BRING BACK HIGH ORDER BIT
1424 005344  005725                             TST     (R5)+           ;BUMP RESULT POINTER
1425 005346  020527  000044'                    CMP     R5,#FEXPB       ;DONE!
1426 005352  103732                             BLO     10$             ;NOPE
1427 005354  005726                             TST     (SP)+           ;YES, KILL BIT MASK
1428 005356  032745  000001                     BIT     #1,-(R5)        ;NEED TO ROUND NUMBER?
1429 005362  001405                             BEQ     11$             ;NO
1430 005364  062715  000001                     ADD     #1,@R5          ;YES, ROUND UP NUMBER
1431 005370  005545                             ADC     -(R5)           ;WITH CARRY
1432 005372  005545                             ADC     -(R5)
1433 005374  005545                             ADC     -(R5)
1434 005376  005737  000034'            11$:    TST     FMANB           ;RESULT POSITIVE?
1435 005402  002013                             BGE     5$              ;YES, GOOD
1436 005404  012700  000004                     MOV     #4,R0           ;4 WORDS TO NEGATE
1437 005410  012701  000044'                    MOV     #FEXPB,R1       ;POINT PAST RESULT
1438 005414  000241                             CLC                     ;CLEAR CARRY
1439
1440 005416  005541                     4$:     ADC     -(R1)           ;ADD THE CARRY
1441 005420  103401                             BCS     6$              ;GENERATED CARRY
1442 005422  005411                             NEG     @R1             ;CORREST SIGN OF RESULT
1443 005424                             6$:     SOB     R0,4$           ;LOOP ONCE PER WORD
1444 005430  005441                             NEG     -(R1)           ;CHANGE SIGN OF NUMBER
1445
1446 005432                             5$:     CALL    FNORMB          ;NORMALIZE B
1447 005436                                     CALL    FPACKB          ;CONVERT TO NUMBER
1448 005442                                     RETURN                  ;AND RETURN
1449
1450
1452
1453
```

```
APLEM   MACRO M1113  08-FEB-80 09:02  PAGE 69
FPCODE           FAKE THE HARDWARE FPP

      1          000001                              .END
```

```
APLEM   MACRO M1113  08-FEB-80 09:02  PAGE 69-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R  025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R  025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R  025
AC5    =%000005          CHRPDL= 000271    C.CZ   = 001000   D.EQ   = 000000    ERRPTR  004770R  025
ALLINT  003274R  025     CHRPLU= 000055    C.DIG  = 000040   D.EXPD= 000002    ERR0    004774R  025
ARYRET  003276R  025     CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R  025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FMT  = 000004   E$CT    005056R  025
A.ALEN= 000004           CHRQUC= 000113    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R  025
A.ARG  = 000040          CHRRBR= 000047    C.FRC  = 000001   D.FPUT= 000005    FAC0    000132RG 026
A.ASB  = 000200          CHRRPR= 000042    C.IDEN= 000010    D.GE   = 000000   FAC1    000142RG 026
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GT   = 000000   FAC2    000152RG 026
A.CON  = 000020          CHRSMO= 000112    C.NOBF= 000400    D.IBM  = 000002   FAC3    000162RG 026
A.DIMS= 000002           CHRSPA= 000040    C.NUM  = 000020   D.IND  = 000002   FAC4    000172RG 026
A.DIM1= 000006           CHRUPA= 000131    C.OP   = 000004   D.INNR= 000013   FAC5    000202RG 026
A.NUM  = 000006          CHR.BS= 000010    C.OPN  = 010000   D.INTR= 000006   FA.APD= 000100
A.PTR  = 000000          CHR.CC= 000003    C.RDLN= 177776    D.LE   = 000000   FA.CRE= 000010
A.TMP  = 000010          CHR.CR= 000015    C.REWT= 002000    D.LOG  = 000001   FA.DLK= 001000
BADDIR= 104463           CHR.CZ= 000032    C.RND  = 000010   D.LT   = 000000   FA.ENB= 100000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.MAX  = 000000   FA.EXC= 002000
BLZBUB  005116R  025     CHR.LF= 000012    C.SPFX= 000100    D.MEM  = 000002   FA.EXT= 000004
BRSELF  005046R  025     CHR.LT= 000043    C.SPGL= 000200    D.MIN  = 000000   FA.NSP= 000100
CCNT    005072R  025     CHR.SI= 000017    C.SPLX= 000001    D.MUL  = 000000   FA.NSR= 000200
CHANEX  003060R  025     CHR.SL= 000057    C.SPRS= 000002    D.NE   = 000000   FA.POS= 010000
CHAN00  000256R  025     CHR.SO= 000016    C.SRC  = 177774   D.NN   = 000000   FA.RD  = 000001
CHAN01  000330R  025     CHR.TB= 000011    C.SRCP= 000024    D.NR   = 000000   FA.RWD= 004000
CHAN02  000512R  025     CHR.UL= 000106    C.TT   = 000004   D.OR   = 000000   FA.SEQ= 040000
CHAN03  000674R  025     CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FA.SHR= 000040
CHAN04  001056R  025     CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FA.TMP= 000020
CHAN05  001240R  025     CHR..S= 000163    DEBUG  = 000000   D.PWR  = 000001   FA.WCK= 020000
CHAN06  001422R  025     CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FA.WRT= 000002
CHAN07  001604R  025     CHR..U= 000165    DEVIL   005114R  025 D.QUAD= 000000  FDESC   000120R  026
CHAN08  001766R  025     CHR..Z= 000172    DEVNTF= 104470    D.RES  = 000000   FDONE   004024R  020
CHAN09  002150R  025     CHR..0= 000060    DIGITS  000006R  025 D.RHO  = 000000  FDONE1  004066R  020
CHAN10  002332R  025     CHR..1= 000061    DIMAX  = 000020   D.ROT  = 000003   FDST    000052R  026
CHAN11  002514R  025     CH.AND= 000001    DIMBLK  003112R  025 D.SCON= 000006
CHAN12  002676R  025     CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006
CHAN13  000022R  023     CMDMAX= 000010    DIMTYP  003116R  025 D.SUB  = 000000
CHAN14  000204R  023     CNTSIZ  005130R  025 DIRFUL= 104466    D.SUBS= 000006
CHNOPN= 104471           COMBIN  005076R  025 DOMERR= 104417    D.TAKE= 000002
CHRAND= 000051           COMCNT  005126R  025 DPTERR= 104416    D.TRAN= 000004
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R  025 D.UNON= 000006
CHRBKR= 000133           COMPTR  005122R  025 D.AND  = 000000   D.$MAX= 000000
CHRBOL= 000200           COMRTN  003156R  025 D.BXCT= 000007    D.$MIN= 000000
CHRBOX= 000114           COMSKP  005124R  025 D.BXIO= 000007    D.$MUL= 000000
CHRCMA= 000054           CORPTR  005120R  025 D.BXNL= 000007    D.$PLU= 000000
CHRCOL= 000076           CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000
CHRDEL= 000107           CS.BOO= 000010    D.BXPW= 000007    D.$SUB= 000000
CHRDIV= 000053           CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001
CHRDLR= 000176           CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002
CHRDLT= 000110           CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003
CHRDOT= 000056           CTEPS   000152R  025 D.COMP= 000001    EDICHR  005063R  025 FD.BLK= 000010
CHREP  = 000105          CTMAX   000212R  025 D.CONT= 000006    EDICNT  005064R  025 FD.CCL= 000002
CHRERR= 104431           CTRMAX  000172R  025 D.CTLM= 000000    EDIFLG  005062R  025 FD.COM= 020000
CHRFIL= 000177           C.ALOC= 000162    D.DEAL= 000003    EDIT   = 000003   FD.CR  = 000002
CHRFRO= 000135           C.BLOK= 000064    D.DECD= 000002    EIS    = 000000   FD.DIR= 000010
                                                                               FD.FTN= 000001
                                                                               FD.F11= 040000
                                                                               FD.INS= 000010
                                                                               FD.ISP= 002000
                                                                               FD.MNT= 100000
                                                                               FD.OSP= 004000
                                                                               FD.PLC= 000004
                                                                               FD.PRN= 000004
                                                                               FD.PSE= 010000
                                                                               FD.RAH= 000001
                                                                               FD.RAN= 000002
                                                                               FD.REC= 000001
                                                                               FD.RWM= 000001
                                                                               FD.SDI= 000020
                                                                               FD.SQD= 000040
                                                                               FD.TTY= 000004
```

APLEM MACRO M1113 08-FEB-80 09:02 PAGE 69-2
SYMBOL TABLE

```
FD.WBH= 000002          FSAVIT  000716R 020 F.FNAM= 000110          IASH    000202RG 020 LINC    005102R 025
FEXPA   000030R 026 FSET    003402R 020 F.FNB = 000102          IASHC   000236RG 020 LINE    005016R 025
FEXPB   000044R 026 FSETA   003370R 020 F.FNDF= 000010          IASHX   000274R 020 LININC  005026R 025
FFLAGS  000124R 026 FSETB   003376R 020 F.FTYP= 000116          IC    = 000001        LISIZE= 000020
FF.CHR= 000005          FSETFL  001532R 020 F.FUN = 000010          ICHAN   005050R 025 LITOP   005072R 025
FF.NV = 000003          FSGNA   000016R 026 F.FVER= 000120          IDIV    000510RG 020 LNLNG = 104541
FF.POE= 000002          FSGNB   000032R 026 F.HEAD= 000006          IFLAGS  000000R 026 LOCLCT  005036R 025
FF.RWD= 000001          FSGNX   005120R 020 F.HIBK= 000004          ILFNAM= 104464          LOWCOR  000002R 025
FF.RWF= 000006          FSIZE   000126R 026 F.LBL = 000020          ILLCHN= 104540          LSTERP  004766R 025
FF.SPC= 000004          FSRC    000046R 026 F.LINE= 000012          ILLSW = 104565          LWIDTH  000070R 025
FGETIT  001434R 020 FSTAT   000056R 026 F.LOC = 000004          IMUL    000376RG 020 L.BITS= 000003
FHEADR  005012R 025 FSTATS  000060R 026 F.LSTC= 000004          IN    = 000010        L.CNT = 000002
FHRET   005040R 025 FS.C  = 000001          F.LUN = 000042          INDERR= 104410          L.PRNT= 000001
FILNTF= 104467          FS.DBL= 000200          F.MBCT= 000054          INNTMP  003250R 025 L.PTR = 000000
FIS   = 000000          FS.LNG= 000100          F.MBC1= 000055          INNTM2  005016R 025 L.PTR1= 000004
FISCON= 046000          FS.N  = 000010          F.MBFG= 000056          INPTR   000064R 025 L.SUBS= 000002
FISIZE  000130R 026 FS.V  = 000002          F.MON = 000002          INSNAM= 104462          MACHIN= 000106
FIXPTR  003074R 025 FS.Z  = 000004          F.NIL = 000001          INZBIT  000350R 020 MAXCHN= 000014
FLAGS   004764R 025 FTMP    000062R 026 F.NRBD= 000024          INZBI1  000362R 020 MONBLK  003120R 025
FLCBUF  003250R 025 FUNDER  004002R 020 F.NREC= 000030          INZBI2  000330R 020 MONCVT  003132R 025
FLINE   005014R 025 FUZZ    000142R 025 F.OPN = 000010          IOERR = 104477          MONLEN  003130R 025
FLOEXP  003270R 025 FVALA   000016R 026 F.OVBS= 000030          IREG    000010R 026 MONPTR  003122R 025
FLONEG  003272R 025 FVALB   000032R 026 F.PREV= 000000          IREG0   000002R 026 MONSFT  003126R 025
FLSWTB  000222R 025 FVSET   002556R 020 F.PRNT= 100000          IREG1   000004R 026 MONTYP  003124R 025
FLTLEN= 000010          FXPTR1  003076R 025 F.PROT= 010000          IRETRN  000006R 026 MONVAL  003134R 025
FMANA   000020R 026 F$EVAL= 000100          F.RACC= 000016          IRETR1  000014R 026 M.ABS = 000000
FMANB   000034R 026 F$EVI = 000200          F.RATT= 000001          ISETUP  000000R 020 M.BOL = 000200
FMODE0  001202R 020 F$EXEC= 000400          F.RCNM= 000034          ISRC    000012R 026 M.BXAV= 000007
FMODE1  001226R 020 F$FUNC= 001000          F.RCTL= 000017          IV    = 000002        M.BXCR= 000007
FMODE2  001234R 020 F.ACTL= 000076          F.READ= 004000          IXOR    000162RG 020 M.BXCT= 000007
FMODE3  001320R 020 F.ALOC= 000040          F.REDF= 000040          IZ    = 000004        M.BXEX= 000007
FMODE4  001332R 020 F.BBFS= 000062          F.RES = 000040          I.CNT = 177776        M.BXFX= 000007
FMODE5  001372R 020 F.BDB = 000070          F.RET = 177772          I.DIM = 177772        M.BXIO= 000007
FMODE6  001404R 020 F.BGBC= 000057          F.RSIZ= 000002          I.LEN = 000012        M.BXLC= 000003
FMODE7  001420R 020 F.BITS= 177774          F.RTYP= 000000          I.OERR= 000062        M.BXNC= 000007
FMOV    000116R 026 F.BKDN= 000026          F.R2  = 000014          I.OLD = 177770        M.BXNL= 000007
FMTBL   001162R 020 F.BKDS= 000020          F.R3  = 000016          I.PTR = 177774        M.BXPP= 000007
FNINSI  005044R 025 F.BKEF= 000050          F.R4  = 000020          I.STR = 177766        M.BXPW= 000007
FNORM   003536R 020 F.BKP1= 000051          F.SEQN= 000100          LABERR= 104406        M.BXRL= 000007
FNORMA  003524R 020 F.BKST= 000024          F.SPDV= 000072          LABLPT  005042R 025 M.BXTS= 000007
FNORMB  003532R 020 F.BKVB= 000064          F.SPUN= 000074          LARGBL  003162R 025 M.BXWA= 000007
FORIGN  000032R 025 F.BR  = 000020          F.STBK= 000036          LARGCV  003176R 025 M.CEI = 177777
FOVER   004014R 020 F.CHR = 000075          F.STK = 000022          LARGFL  003160R 025 M.CHR = 000202
FO.APD= 000106          F.CNTG= 000034          F.STOP= 000004          LARGLE  003174R 025 M.CMT = 000206
FO.MFY= 000002          F.CODE= 000010          F.STP = 000001          LARGLN  003166R 025 M.DOUT= 000006
FO.RD = 000001          F.DFNB= 000046          F.SUSP= 000002          LARGPT  003164R 025 M.DQAD= 000006
FO.UPD= 000006          F.DSPT= 000044          F.TRC = 000002          LARGSF  003172R 025 M.ENCD= 000004
FO.WRT= 000016          F.DVNM= 000134          F.UNIT= 000136          LARGTY  003170R 025 M.EXE = 000000
FPACK   003660R 020 F.EFBK= 000010          F.URBD= 000020          LARGVA  003200R 025 M.EXP = 000001
FPACKA  003646R 020 F.EFN = 000050          F.VAL = 177776          LBINC   005100R 025 M.FACT= 000001
FPACKB  003654R 020 F.END = 002000          F.VBN = 000064          LBUFF   003316R 025 M.FIL = 000177
FPMODE  000042R 025 F.EOBB= 000032          F.VBSZ= 000060          LBUFLN= 000621        M.FLR = 177777
FPP   = 000000          F.ERR = 000052          GNOME   003102R 025 LCOUNT  005106R 025 M.FMT = 000004
FPPLEN= 000004          F.FACC= 000043          HASHC   005136R 025 LENERR= 104412        M.GDUP= 000003
FPTR    001122R 020 F.FFBY= 000014          HASHTB  005140R 025 LENTAB  000054R 025 M.GDWN= 000003
FREBLK  007400R 025 F.FLAG= 000002          HSHMSK= 177601          LEXPTR  003100R 025 M.GETI= 000006
FRTN    000122R 026 F.FLB = 000100          HSHSIZ= 000200          LFTOP   005066R 025 M.IBM = 000003
```

APLEM MACRO M1113 08-FEB-80 09:02 PAGE 69-3
SYMBOL TABLE

```
M.IDN = 000000          ND.$PN= 001400          PATCH   007344R  025 R.VAR = 000002          S.FTYP= 000002
M.INSP= 000006          NEGSGN  005063R  025   PRECSN  000052R  025 R6    =%000006          S.LEN = 000002
M.IOTA= 000000          NINPT1  003302R  025   RANDAC  003064R  025 R7    =%000007          S.LINK= 177774
M.LIT = 000205          NINPT2  003300R  025   RANDY   003062R  025 SI    =%000005          S.NFEN= 000020
M.LN  = 000001          NINSTK  003304R  025   RANERR= 104411          SIDAMN= 104433          S.TYPE= 140000
M.MMD = 000010          NINTMP  003310R  025   RARGBL  003204R  025 SIFRAM  003314R  025 S.VAL = 000000
M.NEG = 000000          NINTYP  003306R  025   RARGCV  003220R  025 SISTAK  006350R  025 TKDIMS  005072R  025
M.NOT = 000000          NOTAVL= 104472          RARGFL  003202R  025 SITOP   006346R  025 TKFRAM  005066R  025
M.NUB = 000006          NOTOPN= 104473          RARGLE  003216R  025 SITP    007144R  025 TKLEFT  005070R  025
M.NUM = 000204          NPRT1   004762R  025   RARGLN  003210R  025 SIWARN= 000200          TMODE   000403R  023
M.OUTR= 000012          NULL    007364R  025   RARGPT  003206R  025 SIZTAB  000044R  025 TMPTAG= 000166
M.PITM= 000000          NUMCHN= 000020          RARGSF  003214R  025 SI.BIT= 177774          TTYBUF  004140R  025
M.PRDS= 000006          NUMDFL  003230R  025   RARGTY  003212R  025 SI.LEN= 177772          TTYLEN= 000622
M.QQAD= 000000          NUMEFL  003233R  025   RARGVA  003222R  025 SI.RET= 177772          TURKEY= 104436
M.QUAD= 000000          NUMESN  003236R  025   RBASE   005074R  025 SI.VAL= 177776          TYPE  = 000166
M.RAND= 000012          NUMFLG  003230R  025   RCOUNR  005112R  025 SKPSIZ  005134R  025 T.BOL = 020000
M.RAVL= 000000          NUMLEN  000062R  025   RCOUNT  005110R  025 SPHOLD  000004R  024 T.CON = 004000
M.RCIP= 000000          NUMNEG  003232R  025   REDCNT  005070R  025 SPINIT  000002R  024 T.DIM = 000100
M.RDCT= 000011          NUMQFL  003234R  025   REDIGN  005066R  025 STKINC= 000774          T.FUN = 010000
M.REVS= 000003          NUMRIG  003231R  025   REDOFF  005072R  025 STOP  = 104420          T.LIST= 000400
M.ROLL= 000000          NUMWID  003226R  025   REENTR  003060R  025 SUBCNT  003104R  025 T.LIT = 000020
M.SATA= 000001          NUMZFL  003235R  025   REG0    000076R  026 SUBDCT  003106R  025 T.MON = 000010
M.SCAN= 000011          NYIERR= 104432          REG1    000100R  026 SUBPTR  003110R  025 T.NIL = 000004
M.SHAP= 000000          N.BLSZ= 000036          REG2    000102R  026 SYMB1   005344R  025 T.NUM = 002000
M.SIGN= 000000          N.DESC= 000020          REG3    000104R  026 SYMF    005340R  025 T.OPFN= 000001
M.SUBR= 000001          N.DEV = 000032          REG4    000106R  026 SYMPT1  005346R  025 T.OTHR= 040000
M.SYM = 000203          N.DEVN= 000034          REG5    000110R  026 SYMTAB  005352R  025 T.SOP = 000002
M.SYN = 000201          N.DID = 000024          REG6    000112R  026 SYMTBF= 104404          T.SPOP= 000040
M.TRAN= 000012          N.DVNM= 000032          REG7    000114R  026 SYMTOP  005350R  025 T.TMP = 000200
M.$ABS= 000000          N.EXT = 000014          RESBIT  003142R  025 SYMTP   006272R  025 T.VAL = 100000
M.$CEI= 000000          N.FID = 000000          RESCVT  003152R  025 SYMWRN= 000054          T.VAR = 001000
M.$FLR= 000000          N.FNAM= 000006          RESERR= 104440          SYNERR= 104407          UPPER = 177400
M.$IDN= 000000          N.FTYP= 000014          RESLEN  003150R  025 SYSERR= 104402          UTNAME  000344R  023
M.$NEG= 000000          N.FVER= 000016          RESPTR  003140R  025 SYSTAT  000000R  024 VALERR= 104413
M.$SIG= 000000          N.NAME= 000006          RESSFT  003146R  025 SY.BAD= 000001          VARBLE  007400R  025
M.1RDT= 000011          N.NEXT= 000022          RESTYP  003144R  025 SY.BRK= 100000          VARBLS  000000R  025
M.1REV= 000003          N.PPN = 000024          RESULT  003136R  025 SY.BS  = 000010          VERNUM  000000R  025
M.1SCN= 000011          N.STAT= 000020          RESVAL  003154R  025 SY.CLR= 000040          VRSION= 000002
NAMBUF  003240R  025    N.UNIT= 000034          RINC    005104R  025 SY.ERP= 000020          WIDTH   000072R  025
NB.DEV= 000200          OBUFLN= 001000          RLTBAS  005066R  025 SY.NOU= 000100          WROUND  000020R  025
NB.DIR= 000100          OCHAN   005052R  025   ROTINC  005070R  025 SY.OPN= 000002          WSDATE  000136R  025
NB.NAM= 000004          OMODE   000402R  023   ROUND   000010R  025 SY.PMT= 001000          WSFULL= 104403
NB.SD1= 000400          OPTMP0  005066R  025   ROVER   003312R  025 SY.SRT= 000400          WSNAME  000074R  025
NB.SD2= 001000          OPTMP1  005070R  025   RSTS  = 000000          SY.SVC= 000004          WSTIME  000132R  025
NB.SNM= 000040          OPTMP2  005072R  025   RSX   = 000115          SY.SVL= 000200          $ABSD   002044RG  020
NB.STP= 000020          OPTMP3  005074R  025   RTOP    005070R  025 S.AB  = 177776          $ABSF   002044RG  020
NB.SVR= 000010          OPTMP4  005076R  025   RT11  = 000000          S.ASB = 100000          $ADDD   003100RG  020
NB.TYP= 000002          OPTMP5  005100R  025   R.BITS= 000003          S.BFHD= 000020          $ADDF   003100RG  020
NB.VER= 000001          OPTMP6  005102R  025   R.FIX = 000001          S.BITS= 000002          $CFCC   001566RG  020
ND.DEV= 000200          OPTMP7  005104R  025   R.LCT = 000002          S.CHR1= 000003          $CLRD   002004RG  020
ND.EXT= 000002          OPTMP8  005106R  025   R.LN  = 000002          S.FAT1= 000016          $CLRF   002004RG  020
ND.NAM= 000004          OPTMP9  005110R  025   R.LNUM= 000010          S.FDB = 000140          $CMPD   003030RG  020
ND.PPN= 000100          OPTM10  005112R  025   R.NEXT= 000006          S.FLB = 040000          $CMPF   003030RG  020
ND.SEX= 000000          OPTM14  005026R  025   R.OFF = 000000          S.FNAM= 000006          $DIVD   003352RG  020
ND.SES= 000020          ORIGIN  000030R  025   R.PREV= 000004          S.FNB = 000036          $DIVF   003352RG  020
ND.SNM= 000000          OUTPTR  000066R  025   R.PTR = 000000          S.FNBW= 000017          $FIS  = ****** G
ND.SNS= 000040          OVROPR= 000000          R.SEQ = 000003          S.FNTY= 000004          $FPP  = ****** G
```

```
APLEM  MACRO M1113  08-FEB-80 09:02  PAGE 69-4
SYMBOL TABLE

$FPPLN= ****** G        $MODF   003144RG  020 $STCDL  002362RG  020 $$ADD2  004614R  020 ..CFLG= 100000
$LDCDF  002576RG  020 $MULD   003116RG  020 $STCFD  002676RG  020 $$ADD3  004622R  020 ..DST = 000400
$LDCFD  002576RG  020 $MULF   003116RG  020 $STCFI  002362RG  020 $$ADD4  004630R  020 ..FDST= 002000
$LDCID  002252RG  020 $NEGD   002072RG  020 $STCFL  002362RG  020 $$ANSW  004572R  020 ..FSRC= 000004
$LDCIF  002252RG  020 $NEGF   002072RG  020 $STD    002126RG  020 $$DIV   005150R  020 ..GDST= 004000
$LDCLD  002252RG  020 $RSTS = ****** G      $STEXP  002210RG  020 $$MUL   004640R  020 ..GSRC= 000010
$LDCLF  002252RG  020 $RSX  = ****** G      $STF    002126RG  020 .BOOLE= 000002       ..IDST= 001000
$LDD    002126RG  020 $RT11 = ****** G      $STFPS  001764RG  020 .CHARA= 000000       ..ISRC= 000002
$LDEXP  002160RG  020 $SETD   001630RG  020 $SUBD   003054RG  020 .DEF. = 000002       ..JDST= 020000
$LDF    002126RG  020 $SETF   001612RG  020 $SUBF   003054RG  020 .FLOAT= 000006       ..PDST= 010000
$LDFPS  001702RG  020 $SETI   001646RG  020 $TSTD   002032RG  020 .FSRCB= ****** G     ..SRC = 000001
$LDF1   002134R   020 $SETL   001664RG  020 $TSTF   002032RG  020 .INTEG= 000004       ...GBL= 000000
$MACHN= ****** G        $STCDF  002676RG  020 $$ADD   004210R  020 .MOLUN  000404R  023 ...TPC= 000020
$MODD   003144RG  020 $STCDI  002362RG  020 $$ADD1  004464R  020

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
MNTRTL  000000  017
CODE    005444  020
PATCH   000000  021
$$FSR1  000000  022
DEF$0   000406  023
DEF$1   000006  024
DEF$2   007400  025
EMULAT  000212  026
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  14209 WORDS  ( 56 PAGES)
DYNAMIC MEMORY:  15534 WORDS  ( 59 PAGES)
ELAPSED TIME:  00:02:33
APLEM,DM1:[305,301]APLEM/-SP=APLRX1,APLDEF,APLEM,END
```

FILEID**APLOP

```
  AAAAAA      PPPPPPPP      LL              000000      PPPPPPPP
  AAAAAA      PPPPPPPP      LL              000000      PPPPPPPP
AA      AA    PP      PP    LL            00      00    PP      PP
AA      AA    PP      PP    LL            00      00    PP      PP
AA      AA    PP      PP    LL            00      00    PP      PP
AA      AA    PP      PP    LL            00      00    PP      PP
AA      AA    PPPPPPPP      LL            00      00    PPPPPPPP
AA      AA    PPPPPPPP      LL            00      00    PPPPPPPP
AAAAAAAAAA    PP            LL            00      00    PP
AAAAAAAAAA    PP            LL            00      00    PP
AA      AA    PP            LL            00      00    PP              ....
AA      AA    PP            LL            00      00    PP              ....
AA      AA    PP            LLLLLLLLLL      000000      PP              ....
AA      AA    PP            LLLLLLLLLL      000000      PP              ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 09:04
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    45-   202    SCALAR FUNCTION         ADDITION
```

APLOP MACRO M1113 08-FEB-80 09:04
TABLE OF CONTENTS

45-	211	SCALAR FUNCTION	ADDITION, FLOATING
45-	227	SCALAR FUNCTION	SUBTRACTION
45-	236	SCALAR FUNCTION	SUBTRACTION, FLOATING
45-	253	SCALAR FUNCTION	MULTIPLY
45-	268	SCALAR FUNCTION	MULTIPLY, FLOATING
45-	287	SCALAR FUNCTION	DIVISION
45-	481	SCALAR FUNCTION	MINIMUM
45-	493	SCALAR FUNCTION	MINIMUM, FLOATING
45-	520	SCALAR FUNCTION	MAXIMUM
45-	532	SCALAR FUNCTION	MAXIMUM, FLOATING
45-	559	SCALAR FUNCTION	RESIDUE
45-	593	SCALAR FUNCTION	RESIDUE, FLOATING
46-	642	SCALAR FUNCTION	AND
46-	656	SCALAR FUNCTION	OR
46-	666	SCALAR FUNCTION	NAND
46-	679	SCALAR FUNCTION	NOR
46-	691	SCALAR FUNCTION	LESS THAN
46-	703	SCALAR FUNCTION	LESS THAN OR EQUAL
46-	716	SCALAR FUNCTION	EQUAL
46-	736	SCALAR FUNCTION	GREATER THAN OR EQUAL
46-	749	SCALAR FUNCTION	GREATER THAN
46-	762	SCALAR FUNCTION	NOT EQUAL
47-	794	SCALAR FUNCTION	IDENTITY
47-	803	SCALAR FUNCTION	IDENTITY, FLOATING
47-	812	SCALAR FUNCTION	FLOOR, FLOATING
47-	859	SCALAR FUNCTION	CEILING, FLOATING
47-	872	SCALAR FUNCTION	NEGATION
47-	881	SCALAR FUNCTION	NEGATION, FLOATING
47-	891	SCALAR FUNCTION	SIGNUM
47-	904	SCALAR FUNCTION	SIGNUM, FLOATING
47-	918	SCALAR FUNCTION	RECIPROCAL
47-	953	SCALAR FUNCTION	PI TIMES
47-	970	SCALAR FUNCTION	ABSOLUTE VALUE
47-	980	SCALAR FUNCTION	ABSOLUTE VALUE, FLOATING
47-	1006	SCALAR FUNCTION	NOT
47-	1035	SCALAR FUNCTION	ROLL
49-	1190	MIXED FUNCTION	SHAPE
49-	1218	MIXED FUNCTION	RESHAPE
50-	1316	MIXED FUNCTION	RAVEL
50-	1332	MIXED FUNCTION	CATENATION/LAMINATION
51-	1599	MIXED FUNCTION	IOTA -- INDEX GENERATOR
68-	4291	MIXED FUNCTION	EXECUTE
73-	5049	MIXED FUNCTION	QUAD INPUT
73-	5082	MIXED FUNCTION	QUAD OUTPUT
73-	5104	MIXED FUNCTION	QUOTE QUAD INPUT
73-	5142	MIXED FUNCTION	QUOTE QUAD OUTPUT

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 1


        1           000000              RT11    =  0          ;NOT RT-11
        2           000000              RSTS    =  0          ;NOT RSTS/E
        3           000115              RSX     = 'M          ;RSX-11/M
        4
        5           000000              EIS     =  0          ;NO EIS
        6           000000              FIS     =  0          ;NO FIS
        7           000000              FPP     =  0          ;NO FPP
        8
        9           000004              FPPLEN  =  4.         ;SINGLE PRECISION
       10
       11           000000              DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  APLOP
                                .IDENT  \V02.03\

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                                ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                                ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                                ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                                ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                                ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                                ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                                ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                                ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                                ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                                ;       INTACT!
                                ;--
000120                          LASTOP  =120                    ;MAXIMUM NUMBER OF OPERATORS!!!
000030                          SCALOP  =30                     ;MAXIMUM NUMBER OF SCALAR OPERATORS!!!


                                ;THESE MACRO DEFINITIONS ARE USED FOR THE RESIDENT OP'S AND OTHER
                                ;VARIOUS SUBROUTINES.  THE MACRO 'PRMRES' IS USED TO PREFIX PERMENENTLY
                                ;RESIDENT CODE WHICH IS FOLLED BY A '.IFTF' AND 'LIST'.
                                ;THE 'LIST' MACRO IS USED TO END OVERLAY/RESIDENT SECTIONS AND MUST
                                ;BE USED AFTER EVERY 'MOPINS', 'DOPINS', OR 'PRMRES' SECTION.  IF
                                ;ANY MORE OP'S ARE ADDED, BE SURE TO INCLUDE THEM IN THE
                                ;RESIDENT/OVERLAY LIST IN THE COMMON FILE.

                                .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP MACRO!
                                .IF GT NUM
                                    BITSX=BITS+0
                                    BITSY=BOZO+0
                                    .PSECT BOPTB
                                    .=BOPTAB+<NUM/2>
                                        .BYTE BITSX!T.OPFN
                                    .PSECT DYDBTB
                                    .=DYDBTL+<NUM/2>
                                        .BYTE BITSY
                                    .ENDC
                                .PSECT CODE
                                $$$$$$=.
                                .PSECT DYDTB
                                .=DYDTAB+NUM
                                .IF EQ D.'NAME
                                        .WORD D$'NAME
                                        .PSECT CODE
                                        .LIST
                                D$'NAME::
                                        .NLIST
                                .IFF
                                    .IF NB NOTME
                                        .WORD   NOTME
                                        .GLOBL  NOTME
                                    .IFF
                                        .GLOBL  D$'NAME
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 44-1
DEFINITIONS     ERRORS

                                                .WORD   D$'NAME
                                            .ENDC
                                                .PSECT  CODE
                                        .IFT
                                        .ENDM

                                        .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                                        .IIF NDF M.'NAME,       .ERROR  ;D.'NAME IS NOT DEFINED BY OVROP MACRO!
                                        .IF GT NUM
                                            BITSX=BITS+0
                                            BITSY=BOZO+0
                                            .PSECT BOPTB
                                            .=BOPTAB+<NUM/2>
                                                .BYTE BITSX!T.OPFN
                                            .PSECT MONBTB
                                            .=MONBTL+<NUM/2>
                                                .BYTE BITSY
                                            .ENDC
                                        .PSECT CODE
                                        $$$$$$=.
                                        .PSECT MONTB
                                        .=MONTAB+NUM
                                        .IF EQ M.'NAME
                                            .IF NE NUM
                                                .WORD M$'NAME
                                                .PSECT CODE
                                                .LIST
                                        M$'NAME::
                                                .NLIST
                                            .ENDC
                                        .IFF
                                            .IF NE NUM
                                                .IF NB NOTME
                                                    .WORD   NOTME
                                                    .GLOBL  NOTME
                                                .IFF
                                                    .WORD   M$'NAME
                                                    .GLOBL  M$'NAME
                                                .ENDC
                                                .PSECT  CODE
                                            .ENDC
                                        .IFT
                                        .ENDM

                                        .MACRO LIST
                                                .ENDC
                                        .ENDM

                                        .MACRO PRMRFS
                                        .IF EQ 0
                                        .ENDM

                                                        ;********** ASSEMBLING THE FILE!!!!!


                                                        ;********** ASSEMBLING THE FILE!!!!!
160 007400                              PRMRES
```

```
                                ;TABLE OF MONADIC OPERATORS
                                .PSECT  MONTB
000000                          MONTAB::
                                        WFILL   SCALOP,SYNER
000060                                  WFILL   LASTOP,SYNER

                                ;TABLE OF DYADIC OPERATORS
                                .PSECT  DYDTB
000000                          DYDTAB::
                                        WFILL   SCALOP,SYNER
000060                                  WFILL   LASTOP,SYNER

                                ;OPERATOR DESCRIPTION BITS (BYTE TABLE!)
                                .PSECT  BOPTB
000000                          BOPTAB::
                                        BFILL LASTOP,0

                                ;TABLE OF DYADIC OPERATOR BITS (BYTE TABLE!)
                                .PSECT  DYDBTB
000000                          DYDBTL::
                                        BFILL   SCALOP,0

                                ;TABLE OF MONADIC OPERATOR BITS (BYTE TABLE!)
                                .PSECT  MONBTB
000000                          MONBT ::
                                        BFILL   SCALOP,0

                                        LIST
```

```
                                DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
000001      003                         .BYTE BITSX!T.OPFN
000001      244                         .BYTE BITSY
000062  000000'                         .WORD D$PLUS
000000                          D$PLUS::
                                .SBTTL  SCALAR FUNCTION          ADDITION

000000  013737  003200' 003154'         MOV     LARGVA,RESVAL    ;GET LEFT ARG
000006  063737  003222' 003154'         ADD     RARGVA,RESVAL    ;ADD RIGHT ARG
000014                                  RETURN
000016                                  LIST

000016                          DOPINS  $PLU,-2
000056  000016'                         .WORD D$$PLU
000016                          D$$PLU::
                                .SBTTL  SCALAR FUNCTION          ADDITION, FLOATING

000016                                  STF     AC1,AC0          ;GET LEFT ARG
000024                                  ADDF    AC2,AC0          ;ADD RIGHT ARG
000032                                  RETURN
000034                                  LIST

000034                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
000002      003                         .BYTE BITSX!T.OPFN
000002      044                         .BYTE BITSY
000064  000034'                         .WORD D$SUB
000034                          D$SUB::
                                .SBTTL  SCALAR FUNCTION          SUBTRACTION

000034  013737  003200' 003154'         MOV     LARGVA,RESVAL    ;GET LEFT ARG
000042  163737  003222' 003154'         SUB     RARGVA,RESVAL    ;SUBTRACT RIGHT ARG
000050                                  RETURN
000052                                  LIST

000052                          DOPINS  $SUB,-4
000054  000052'                         .WORD D$$SUB
000052                          D$$SUB::
                                .SBTTL  SCALAR FUNCTION          SUBTRACTION, FLOATING

000052                                  STF     AC1,AC0          ;GET LEFT ARG
000060                                  SUBF    AC2,AC0          ;SUBTRACT RIGHT ARG
000066                                  RETURN
000070                                  LIST


000070                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
000003      003                         .BYTE BITSX!T.OPFN
000003      354                         .BYTE BITSY
000066  000070'                         .WORD D$MUL
000070                          D$MUL::
                                .SBTTL  SCALAR FUNCTION          MULTIPLY

000070                                  PUSH    R1               ;SAVE WORKING REGISTER
000072  013701  003200'                 MOV     LARGVA,R1        ;GET LEFT ARG
000076                                  MUL     RARGVA,R1        ;TIMES RIGHT ARG
000112  010137  003154'                 MOV     R1,RESVAL        ;STORE RESULT
000116                                  POP     R1               ;RESTORE R1 (CARRY BIT THE SAME)
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 45-1
SCALAR FUNCTION          MULTIPLY

    260 000120  103001                      BCC     1$              ;FITS IN 16 BITS
    261 000122  000262                      SEV                     ;DOESN'T FIX, OVERFLOW
    262 000124                      1$:     RETURN
    264 000126                              LIST
    265
    266
    267 000126                      DOPINS  $MUL,-6
        000052  000126'                     .WORD D$$MUL
        000126                      D$$MUL::
    268                             .SBTTL  SCALAR FUNCTION         MULTIPLY, FLOATING
    269
    276 000126                              STF     AC1,AC0         ;GET LEFT ARG
    277 000134                              MULF    AC2,AC0         ;MULTIPLY BY RIGHT ARG
    278 000142                              CFCC                    ;OVERFLOW?
    279 000146  102001                      BVC     1$              ;NO
    280 000150  104417                      DOMERR                  ;YES
    282 000152                      1$:     RETURN
    284 000154                              LIST
    285
    286 000154                      DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
        000004     003                      .BYTE BITSX!T.OPFN
        000004     156                      .BYTE BITSY
        000070  000154'                     .WORD D$DIV
        000154                      D$DIV::
    287                             .SBTTL  SCALAR FUNCTION         DIVISION
    288
    306 000154                              TSTF    AC2             ;DIVISION BY ZERO?
    307 000162                              CFCC                    ;GET CONDITION CODES
    308 000166  001013                      BNE     1$              ;NON-ZERO, GOOD
    309 000170                              TSTF    AC1             ;DIVIDE INTO ZERO?
    310 000176                              CFCC                    ;GET CODES
    311 000202  001017                      BNE     3$              ;NO, DIVIDE BY 0
    312 000204                              LDF     #^F1.0,AC0      ;GET A 1.0
    313 000214  000411                      BR      2$              ;DONE
    314 000216                      1$:     STF     AC1,AC0         ;GET LEFT ARG
    315 000224                              DIVF    AC2,AC0         ;DIVIDE BY RIGHT ARG
    316 000232                              CFCC                    ;CHECK FOR OVER/UNDER FLOW
    317 000236  102401                      BVS     3$              ;OOPS!
    319 000240                      2$:     RETURN
    320
    321 000242  104417              3$:     DOMERR                  ;DOMAIN ERROR
    323 000244                              LIST
    324
    325
    326 000244                      DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
        000010     003                      .BYTE BITSX!T.OPFN
        000010     156                      .BYTE BITSY
        000100  000000G                     .WORD   D$PWR
    392 000244                              LIST
    393
    394
    395 000244                      DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
        000026     003                      .BYTE BITSX!T.OPFN
        000026     016                      .BYTE BITSY
        000134  000000G                     .WORD   D$LOG
    417 000244                              LIST
    418
```

```
 419
 420 000244                              DOPINS  CIRC,52,T.SOP,.FLOAT
     000025      003                             .BYTE BITSX!T.OPFN
     000025      006                             .BYTE BITSY
     000132  000000G                             .WORD   D$CIRC
 477 000244                                      LIST
 478
 479
 480 000244                              DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
     000006      003                             .BYTE BITSX!T.OPFN
     000006      214                             .BYTE BITSY
     000074  000244'                             .WORD D$MIN
     000244                              D$MIN::
 481                                     .SBTTL  SCALAR FUNCTION          MINIMUM
 482
 483 000244                                      PUSH    LARGVA           ;ASSUME LEFT ARG
 484 000250  021637  003222'                     CMP     @SP,RARGVA       ;WERE WE RIGHT?
 485 000254  002402                              BLT     1$               ;OF COURSE
 486 000256  013716  003222'                     MOV     RARGVA,@SP       ;...NOT
 487 000262                              1$:     POP     RESVAL           ;SAVE THE RESULT
 488 000266                                      RETURN
 490 000270                                      LIST
 491
 492 000270                              DOPINS  $MIN,-14
     000044  000270'                             .WORD D$$MIN
     000270                              D$$MIN::
 493                                     .SBTTL  SCALAR FUNCTION          MINIMUM, FLOATING
 494
 507 000270                                      CMPF    AC1,AC2          ;COMPARE ARGS
 508 000276                                      CFCC                     ;GET CODES
 509 000302  003404                              BLE     1$               ;A<B
 510 000304                                      STF     AC2,AC0          ;RIGHT ARG IS SMALLER
 511 000312  000403                              BR      3$               ;DONE
 512 000314                              1$:     STF     AC1,AC0          ;LEFT ARG IS SMALLER
 514 000322                              3$:     RETURN
 516 000324                                      LIST
 517
 518
 519 000324                              DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
     000007      003                             .BYTE BITSX!T.OPFN
     000007      214                             .BYTE BITSY
     000076  000324'                             .WORD D$MAX
     000324                              D$MAX::
 520                                     .SBTTL  SCALAR FUNCTION          MAXIMUM
 521
 522 000324                                      PUSH    LARGVA           ;ASSUME LEFT ARG
 523 000330  021637  003222'                     CMP     @SP,RARGVA       ;WERE WE RIGHT?
 524 000334  003002                              BGT     1$               ;OF COURSE
 525 000336  013716  003222'                     MOV     RARGVA,@SP       ;...NOT
 526 000342                              1$:     POP     RESVAL           ;SAVE THE RESULT
 527 000346                                      RETURN
 529 000350                                      LIST
 530
 531 000350                              DOPINS  $MAX,-16
     000042  000350'                             .WORD D$$MAX
     000350                              D$$MAX::
 532                                     .SBTTL  SCALAR FUNCTION          MAXIMUM, FLOATING
```

```
                                                CMPF    AC1,AC2         ;COMPARE ARGS
                                                CFCC                    ;GET CODES
                                                BGE     1$              ;USE A
                                                STF     AC2,AC0         ;RIGHT ARG IS LARGER
                                                BR      3$              ;DONE
                                        1$:     STF     AC1,AC0         ;LEFT ARG IS LARGER
                                        3$:     RETURN
                                                LIST


                                        DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
                                                .BYTE BITSX!T.OPFN
                                                .BYTE BITSY
                                                .WORD D$RES
                                        D$RES::
                                        .SBTTL  SCALAR FUNCTION         RESIDUE

                                                TST     LARGVA          ;Is the Left argument Zero?
                                                BNE     2$              ;No
                                        1$:     MOV     RARGVA,RESVAL   ;Yes, Use Right argument for Result
                                                RETURN                  ;Finished
                                        2$:     TST     RARGVA          ;Is the Right argument Zero?
                                                BEQ     1$              ;Yes

                                                PUSH    <R1,R0>         ;No, Save some Registers
                                                CLR     R0              ;Assume Right argument Positive
                                                MOV     RARGVA,R1       ;Copy the Right argument, Is it Negative?
                                                BGE     3$              ;No, Value is non-negative
                                                DEC     R0              ;Yes, Sign-Extend
                                        3$:     DIV     LARGVA,R0       ;Do the Division,  Right / Left
                                                TST     R1              ;Is the Remainder Zero?
                                                BEQ     5$              ;Yes, Correction not required
                                                TST     RARGVA          ;Is the Right argument positive?
                                                BGE     6$              ;Yes
                                                TST     LARGVA          ;No, Is the Left argument positive?
                                                BLE     5$              ;No
                                        4$:     ADD     LARGVA,R1       ;Yes, correct the Remainder
                                        5$:     MOV     R1,RESVAL       ;Store the Result
                                                POP     <R0,R1>         ;Finished
                                                RETURN

                                        6$:     TST     LARGVA          ;Is the Left argument Negative?
                                                BGE     5$              ;No
                                                BR      4$              ;Yes, Correction required
                                                LIST


                                        DOPINS  $RES,-12
                                                .WORD D$$RES
                                        D$$RES::
                                        .SBTTL  SCALAR FUNCTION         RESIDUE, FLOATING

                                                TSTF    AC1             ;SEE WHAT SIGN OUR MODULO HAS
                                                CFCC
                                                BNE     1$              ;SPECIAL CASE IF MODULO IS ZERO
                                                STF     AC2,AC0         ;IF MODULO=0, REMAINDER IS RIGHT ARG
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 45-4
SCALAR FUNCTION          RESIDUE, FLOATING

    599 000552                                   RETURN                  ;FINISHED ALREADY ?
    600
    601 000554                              1$:  STF     AC1,AC4         ;SAVE LEFT SIDE
    602 000562                                   STF     AC2,AC3         ;NEED TO USE THIS
    603 000570                                   DIVF    AC1,AC3         ;FIND RIGHT/LEFT
    604 000576                                   STF     AC3,AC0         ;COPY RIGHT/LEFT
    605 000604                                   MODF    #^F1.0,AC0      ;AND GRAB THE FRACTION OF THAT DIVISION
    606 000614                                   CMPF    AC0,AC3         ;RESULT < ARG ?
    607 000622                                   CFCC
    608 000626  003404                           BLE     2$              ;<= ARG, NO FIXUP
    609 000630                                   SUBF    #^F1.0,AC1      ;FIXUP FOR NEGATIVE
    610 000640                              2$:  MULF    AC4,AC1         ;NOW MAKE IT A REMAINDER
    611 000646                                   STF     AC2,AC0         ;SAVE REMAINDER AS POSSIBLE RESIDUE
    612 000654                                   SUBF    AC1,AC0         ; B - A * FLOOR(B / A)
    613 000662                                   LDF     AC4,AC1         ;RESTORE AC1
    614 000670                                   RETURN
    616 000672                                   LIST
    617
    618
    619 000672                              DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
        000011      003                          .BYTE BITSX!T.OPFN
        000011      156                          .BYTE BITSY
        000102  000000G                          .WORD   D$COMB
    639 000672                                   LIST
```

```
                                DOPINS  AND,42,T.SOP,,BOOLEAN+CS.IDN+CS.ONE+CS.ASS
000021    003                           .BYTE BITSX!T.OPFN
000021    342                           .BYTE BITSY
000122  000672'                         .WORD D$AND
000672                          D$AND::
                                .SBTTL  SCALAR FUNCTION          AND

000672  105037  003154'               CLRB    RESVAL            ;ASSUME 0
000676  105737  003200'               TSTB    LARGVA            ;CHECK ARG
000702  001405                        BEQ     1$                ;0 RESULT
000704  105737  003222'               TSTB    RARGVA            ;CHECK ARG
000710  001402                        BEQ     1$                ;0 RESULT
000712  105237  003154'               INCB    RESVAL            ;SET TO 1
000716                          1$:     RETURN
000720                                  LIST


000720                          DOPINS  OR,40,T.SOP,,BOOLEAN+CS.IDN+CS.ASS
000020    003                           .BYTE BITSX!T.OPFN
000020    242                           .BYTE BITSY
000120  000720'                         .WORD D$OR
000720                          D$OR::
                                .SBTTL  SCALAR FUNCTION          OR

000720  113737  003200' 003154'       MOVB    LARGVA,RESVAL     ;GET ONE ARG
000726  153737  003222' 003154'       BISB    RARGVA,RESVAL     ;OR IT
000734                                  RETURN
000736                                  LIST


000736                          DOPINS  NN,46,T.SOP,,BOOLEAN
000023    003                           .BYTE BITSX!T.OPFN
000023    002                           .BYTE BITSY
000126  000736'                         .WORD D$NN
000736                          D$NN::
                                .SBTTL  SCALAR FUNCTION          NAND

000736  105037  003154'               CLRB    RESVAL            ;ASSUME 0
000742  105737  003200'               TSTB    LARGVA            ;GET ARG
000746  001403                        BEQ     1$                ;IT'S 0, RETURN 1
000750  105737  003222'               TSTB    RARGVA            ;GET ARG
000754  001002                        BNE     2$                ;IT'S 1, RETURN 0
000756  105237  003154'          1$:  INCB    RESVAL            ;RETURN A 1
000762                          2$:     RETURN
000764                                  LIST


000764                          DOPINS  NR,44,T.SOP,,BOOLEAN
000022    003                           .BYTE BITSX!T.OPFN
000022    002                           .BYTE BITSY
000124  000764'                         .WORD D$NR
000764                          D$NR::
                                .SBTTL  SCALAR FUNCTION          NOR

000764  113737  003200' 003154'       MOVB    LARGVA,RESVAL     ;COPY 1ST ARG
000772  105137  003154'               COMB    RESVAL            ;NOT (1ST ARG)
000776  143737  003222' 003154'       BICB    RARGVA,RESVAL     ;(NOT (1ST ARG)) AND (NOT (2ND ARG))
001004  142737  177776  003154'       BICB    #^C1,RESVAL       ;CLEAR GARBAGE BITS
```

```
                                        RETURN
                                        LIST


                        DOPINS  LT,24,T.LIT+T.SOP,,CHARACTER+CS.IDN
                                .BYTE BITSX!T.OPFN
                                .BYTE BITSY
                                .WORD D$LT
                        D$LT::
                        .SBTTL  SCALAR FUNCTION          LESS THAN

                                CLRB    RESVAL           ;ASSUME 0
                                CALL    @COMRTN          ;COMMON COMPARE RTN
                                BGE 1$
                                INCB    RESVAL           ;IT'S A 1
                        1$:     CLV
                                RETURN
                                LIST

                        DOPINS  LE,26,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
                                .BYTE BITSX!T.OPFN
                                .BYTE BITSY
                                .WORD D$LE
                        D$LE::
                        .SBTTL  SCALAR FUNCTION          LESS THAN OR EQUAL

                                CLRB    RESVAL           ;ASSUME 0
                                CALL    @COMRTN          ;COMMON COMPARE RTN
                                BGT     1$               ;NOT 'LESS THAN OR EQUAL TO'
                                INCB    RESVAL           ;IT'S A 1
                        1$:     CLV
                                RETURN
                                LIST


                        DOPINS  EQ,30,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
                                .BYTE BITSX!T.OPFN
                                .BYTE BITSY
                                .WORD D$EQ
                        D$EQ::
                        .SBTTL  SCALAR FUNCTION          EQUAL
                                CLRB    RESVAL           ;Assume result is zero (i.e. Not equal)
                                TST     LARGTY           ;Is the Left argument character type?
                                BEQ     3$               ;Yes
                                TST     RARGTY           ;No, Is the Right argument character type?
                                BEQ     2$               ;Yes

                        1$:     CALL    @COMRTN          ;Arguments have the same type, Are they equal?
                                BNE     2$               ;Not 'EQUAL'
                                INCB    RESVAL           ;Arguments are Equal
                        2$:     CLV
                                RETURN

                        3$:     TST     RARGTY           ;Is the Right argument character type?
                                BNE     2$               ;No
                                BR      1$               ;Yes
                                LIST
```

```
                                
001126                          DOPINS  GE,36,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
000017    023                           .BYTE BITSX!T.OPFN
000017    140                           .BYTE BITSY
000116  001126'                         .WORD D$GE
001126                          D$GE::
                                .SBTTL  SCALAR FUNCTION          GREATER THAN OR EQUAL
                                
001126  105037  003154'               CLRB    RESVAL            ;ASSUME 0
001132                                CALL    @COMRTN           ;COMMON COMPARE RTN
001136  002402                        BLT     1$                ;Not 'GREATER THAN OR EQUAL''
001140  105237  003154'               INCB    RESVAL            ;SET TO 1
001144  000242                  1$:   CLV
001146                                RETURN
001150                                LIST
                                
                                
001150                          DOPINS  GT,32,T.LIT+T.SOP,,CHARACTER+CS.IDN
000015    023                           .BYTE BITSX!T.OPFN
000015    040                           .BYTE BITSY
000112  001150'                         .WORD D$GT
001150                          D$GT::
                                .SBTTL  SCALAR FUNCTION          GREATER THAN
                                
001150  105037  003154'               CLRB    RESVAL            ;ASSUME 0
001154                                CALL    @COMRTN           ;COMMON COMPARE ROUTINE
001160  003402                        BLE     1$                ;NOT 'GREATER THAN''
001162  105237  003154'               INCB    RESVAL            ;SET TO 1
001166  000242                  1$:   CLV
001170                                RETURN
001172                                LIST
                                
                                
001172                          DOPINS  NE,34,T.LIT+T.SOP,,CHARACTER+CS.IDN
000016    023                           .BYTE BITSX!T.OPFN
000016    040                           .BYTE BITSY
000114  001172'                         .WORD D$NE
001172                          D$NE::
                                .SBTTL  SCALAR FUNCTION          NOT EQUAL
                                
001172  105037  003154'               CLRB    RESVAL            ;Assume the result is zero (i.e. Equal)
001176  005737  003170'               TST     LARGTY            ;Is the Left argument character type?
001202  001412                        BEQ     4$                ;Yes
001204  005737  003212'               TST     RARGTY            ;No, Is Right argument chcaracter type?
001210  001403                        BEQ     2$                ;Yes
                                
001212                          1$:   CALL    @COMRTN           ;Args have the same type, Are they Not equal?
001216  001402                        BEQ     3$                ;Equal
001220  105237  003154'         2$:   INCB    RESVAL            ;Not equal
001224  000242                  3$:   CLV
001226                                RETURN
                                
001230  005737  003212'         4$:   TST     RARGTY            ;Is the Right argument character type?
001234  001371                        BNE     2$                ;No
001236  000765                        BR      1$                ;Yes
001240                                LIST
                                
```

```
                                        MOPINS  FLR,14,T.SOP,,.INTEGER+CS.BOOL+CS.FLOAT,M$IDN
000006      003                                 .BYTE BITSX!T.OPFN
000006      034                                 .BYTE BITSY
000074  001240'                                     .WORD   M$IDN
                                        LIST

                                        MOPINS  CEI,16,T.SOP,,.INTEGER+CS.BOOL+CS.FLOAT,M$IDN
000007      003                                 .BYTE BITSX!T.OPFN
000007      034                                 .BYTE BITSY
000076  001240'                                     .WORD   M$IDN
                                        LIST

                                        MOPINS  IDN,2,T.SOP,,.INTEGER+CS.BOOL
000001      003                                 .BYTE BITSX!T.OPFN
000001      014                                 .BYTE BITSY
000062  001240'                                 .WORD M$IDN
001240                                  M$IDN::
                                        .SBTTL  SCALAR FUNCTION          IDENTITY

001240  013737  003134' 003154'                MOV     MONVAL,RESVAL    ;VERY SIMPLE
001246                                          RETURN
001250                                          LIST


001250                                  MOPINS  $IDN,-2
000056  001250'                                 .WORD M$$IDN
001250                                  M$$IDN::
                                        .SBTTL  SCALAR FUNCTION          IDENTITY, FLOATING

001250                                          STF     AC1,AC0          ;THIS IS EASY!
001256                                          RETURN
001260                                          LIST


001260                                  MOPINS  $FLR,-14
000044  001260'                                 .WORD M$$FLR
001260                                  M$$FLR::
                                        .SBTTL  SCALAR FUNCTION          FLOOR, FLOATING

001260                                          PUSH    R0               ;Get a working register
001262  005000                                  CLR     R0               ;Assume a positive argument
001264                                          TSTF    AC1              ;Is it really positive?
001272                                          CFCC
001276  002002                                  BGE     1$               ;Yes
001300  012700  000010                          MOV     #8.,R0           ;No, point to constants for negative argument
001304                                  1$:     STF     AC1,AC3          ;Copy the argument
001312                                          CALL    TRUFLR           ;Get the true floor in AC1
001316                                          ADDF    #^F1.0,AC1       ;1 + .FL A
001326                                          ABSF    AC1              ;.AB 1 + .FL A
001334                                          MULF    CTEPS(R0),AC1    ;EPS # .AB 1 + .FL A
001344                                          CMPF    CTRMAX(R0),AC1   ;RMAX < EPS#.AB 1+.FL A ?
001354                                          CFCC
001360  002004                                  BGE     2$               ;No
001362                                          LDF     CTRMAX(R0),AC1   ;Yes, Now have RMAX .FL EPS # .AB 1 + .FL A
001372                                  2$:     CMPF    FUZZ,AC1         ;BXCT > RMAX.FL EPS#.AB 1+.FL A ?
001402                                          CFCC
001406  003404                                  BLE     3$               ;No
```

```
 832 001410                                     LDF     FUZZ,AC1        ;Yes, Now have BXCT .CE RMAX .FL EPS#.AB 1+.FL A
 833 001420                             3$:     ADDF    AC3,AC1         ;A + BXCT .CE RMAX .FL EPS # .AB 1 + .FL A
 834 001426                                     CALL    TRUFLR          ;.FL A + BXCT .CE RMAX .FL EPS # .AB 1 + .FL A
 835 001432                                     STF     AC1,AC0         ;Copy the above and call it F
 836 001440                                     SUBF    AC3,AC1         ;F - A
 837 001446                                     CMPF    CTFMAX(R0),AC1  ;Is (F - A) >= RMAX ?
 838 001456                                     CFCC
 839 001462  003012                             BGT     4$              ;No
 840 001464                                     TSTF    AC3             ;Is the argument > 0 ?
 841 001472                                     CFCC
 842 001476  003404                             BLE     4$              ;No
 843 001500                                     SUBF    #^F1.0,AC0      ;Correction for plus argument
 844 001510                             4$:     STF     AC3,AC1         ;Restore argument
 845 001516                                     POP     R0              ;Restore working register
 846 001520                                     RETURN
 847
 848 001522                             TRUFLR: STF     AC1,AC0         ;Copy the argument to the correct register
 849 001530                                     MODF    #^F1.0,AC0      ;Get pseudo floor
 850 001540                                     CFCC                    ;Was the argument negative?
 851 001544  002004                             BGE     1$              ;No
 852 001546                                     SUBF    #^F1.0,AC1      ;Yes, Correction for negative arguments
 853 001556                             1$:     RETURN
 855 001560                                     LIST
 856
 857
 858 001560                             MOPINS  $CEI,-16
     000042  001560'                            .WORD M$$CEI
     001560                             M$$CEI::
 859                                    .SBTTL  SCALAR FUNCTION         CEILING, FLOATING
 860
 861 001560                                     STF     AC1,-(SP)       ;Save the argument
 862 001566                                     NEGF    AC1             ;  - A
 863 001574                                     CALL    M$$FLR          ;  .fl - A
 864 001600                                     NEGF    AC0             ;  - .fl - A
 865 001606                                     LDF     (SP)+,AC1       ;Restore the argument
 866 001614                                     RETURN
 868 001616                                     LIST
 869
 870
 871 001616                             MOPINS  NEG,4,T.SOP,,INTEGER
     000002      003                            .BYTE BITSX!T.OPFN
     000002      004                            .BYTE BITSY
     000064  001616'                            .WORD M$NEG
     001616                             M$NEG::
 872                                    .SBTTL  SCALAR FUNCTION         NEGATION
 873
 874 001616  013737  003134' 003154'             MOV     MONVAL,RESVAL   ;COPY VALUE
 875 001624  005437  003154'                     NEG     RESVAL          ;AND NEGATE IT
 876 001630                                     RETURN
 878 001632                                     LIST
 879
 880 001632                             MOPINS  $NEG,-4
     000054  001632'                            .WORD M$$NEG
     001632                             M$$NEG::
 881                                    .SBTTL  SCALAR FUNCTION         NEGATION, FLOATING
 882
 883 001632                                     STF     AC1,AC0         ;COPY ARG
```

```
                                        NEGF    AC0             ;NO SWEAT
                                        RETURN
                                        LIST


                                MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
                                        .BYTE BITSX!T.OPFN
                                        .BYTE BITSY
                                        .WORD M$SIGN
                                M$SIGN::
                                .SBTTL  SCALAR FUNCTION         SIGNUM

                                        CLR     RESVAL          ;CLEAR (ASSUME 0)
                                        TST     MONVAL          ;TEST ARG
                                        BLE     1$              ;LESS THAN, SIGN NEGATIVE
                                        INC     RESVAL          ;VALUE IS 1
                                1$:     BGE     2$              ;NOT NEGATIVE
                                        DEC     RESVAL          ;VALUE IS -1
                                2$:     RETURN
                                        LIST

                                MOPINS  $SIG,-6
                                        .WORD M$$SIG
                                M$$SIG::
                                .SBTTL  SCALAR FUNCTION         SIGNUM, FLOATING
                                        CLRF    AC0             ;CLEAR (ASSUME 0)
                                        TSTF    AC1             ;TEST ARG
                                        CFCC                    ;GET CODES
                                        BLE     1$              ;LESS THAN, SIGN NEGATIVE
                                        LDF     #^F1.0,AC0      ;VALUE IS 1
                                1$:     BGE     2$              ;NOT NEGATIVE
                                        SUBF    #^F1.0,AC0      ;VALUE IS -1
                                2$:     RETURN
                                        LIST


                                MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
                                        .BYTE BITSX!T.OPFN
                                        .BYTE BITSY
                                        .WORD M$RCIP
                                M$RCIP::
                                .SBTTL  SCALAR FUNCTION         RECIPROCAL

                                        TSTF    AC1             ;TEST ARG
                                        CFCC                    ;GET CODES
                                        BEQ     2$              ;NON-ZERO, GOOD
                                        LDF     #^F1.0,AC0      ;GET 1.0
                                        DIVF    AC1,AC0         ;DIVIDE BY ARG
                                        RETURN
                                2$:     DOMERR                  ;DOMAIN ERROR
                                        LIST

                                MOPINS  EXP,20,T.SOP,,FLOAT
                                        .BYTE BITSX!T.OPFN
                                        .BYTE BITSY
                                            .WORD   M$EXP
                                        LIST
```

```
                                        MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
                                                .BYTE BITSX!T.OPFN
                                                .BYTE BITSY
                                                    .WORD   M$LN
                                                LIST


                                        MOPINS  PITM,52,T.SOP,,FLOAT
                                                .BYTE BITSX!T.OPFN
                                                .BYTE BITSY
                                                .WORD M$PITM
                                        M$PITM::
                                        .SBTTL  SCALAR FUNCTION         PI TIMES

                                                STF     AC1,AC0         ;COPY ARG
                                                MULF    PI,AC0          ;MULTIPLY BY PI
                                                RETURN
                                                LIST

                                        PRMRES
                                                .PSECT  TABLES

000000  040511  007732  121041   PI::    .FLT4 3.14159265358979323846
000006  064302
                                                LIST


                                        MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
                                                .BYTE BITSX!T.OPFN
                                                .BYTE BITSY
                                                .WORD M$ABS
                                        M$ABS::
                                        .SBTTL  SCALAR FUNCTION         ABSOLUTE VALUE

002022  013737  003134' 003154'                MOV     MONVAL,RESVAL   ;COPY VALUE
002030  002002                                BGE     1$              ;POSITIVE NUMBER
002032  005437  003154'                        NEG     RESVAL          ;NO, MAKE IT SO
002036                                  1$:     RETURN
002040                                          LIST


002040                                  MOPINS  $ABS,-12
000046  002040'                                 .WORD M$$ABS
002040                                  M$$ABS::
                                        .SBTTL  SCALAR FUNCTION         ABSOLUTE VALUE, FLOATING

002040                                          STF     AC1,AC0         ;COPY ARG
002046                                          ABSF    AC0             ;GOOD OLD FPP!
002054                                          RETURN
002056                                          LIST


002056                                  MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
000011      003                                 .BYTE BITSX!T.OPFN
000011      016                                 .BYTE BITSY
000102  0000006                                     .WORD   M$FACT
```

```
                                        LIST
                                
                                
                                MOPINS  NOT,50
                                        .BYTE BITSX!T.OPFN
                                        .BYTE BITSY
                                        .WORD M$NOT
                                M$NOT::
                                .SBTTL  SCALAR FUNCTION          NOT
                                
                                        MOV     MONBLK,R2        ;Get pointer to data array block
                                        MOV     MONTYP,R1        ;Get source type
                                        BNE     1$               ;Character type?
                                        DOMERR                   ;Yes, A No No
                                1$:     MOV     @MONCT(R1),MONCVT  ;Get source fetch routine pointer
                                        MOV     #SM.BB,RESCVT      ;Get result store routine pointer
                                        MOV     R2,R0            ;Copy the source pointer
                                        MOV     #.BOOLEAN,R1     ;Result type is Boolean
                                        CALL    GETSM3           ;Get some space for the result
                                        MOV     A.ALEN(R2),R3    ;Get the size of the result
                                        BEQ     3$               ;Is the source a null array?
                                        MOV     (R2),R2          ;No, Point to the source data
                                
                                2$:     CALL    @MONCVT          ;Fetch a source value
                                        MOVB    MONVAL,RESVAL    ;Copy it to the result, [0,1]
                                        NEGB    RESVAL           ;Complement it, [0,-1]
                                        INCB    RESVAL           ;Convert to Boolean, [1,0]
                                        CALL    @RESCVT          ;Store it away
                                        SYSERR
                                        SOB     R3,2$            ;Loop for all the elements
                                
                                3$:     RETURN                   ;All done
                                        LIST
                                
                                
                                MOPINS  ROLL,100
                                        .BYTE BITSX!T.OPFN
                                        .BYTE BITSY
                                        .WORD M$ROLL
                                M$ROLL::
                                .SBTTL  SCALAR FUNCTION          ROLL
                                
                                        MOV     MONBLK,R2        ;POINT TO ARG BLOCK
                                        MOV     MONTYP,R0        ;GET THE SOURCE TYPE
                                        BNE     1$               ;YES
                                        DOMERR                   ;DOMAIN ERROR
                                1$:     MOV     FMONCT(R0),MONCVT ;MONADIC FETCH
                                        MOV     FRESCT(R0),RESCVT ;RESULT STORE ROUTINE
                                        MOV     R2,R0            ;COPY POINTER
                                        CALL    GETSM1           ;GET RESULT CORE
                                        MOV     A.ALEN(R2),R3    ;GET # OF NUMBERS
                                        BEQ     3$               ;DONE IF NONE
                                        MOV     (R2),R2          ;POINT TO ARG DATA
                                
                                2$:     CALL    @MONCVT          ;GET MONADIC SOURCE
                                        STF     AC1,AC0          ;TRANSFER IT
                                        CALL    .RAND            ;RANDOMIZE!
```

```
 1052 002252                                    CALL    @RESCVT         ;SAVE VALUE
 1053 002256  104402                            SYSERR                  ;DIE, SYSTEM ERROR
 1054 002260                                    SOB     R3,2$           ;LOOP TILL DONE
 1055 002264                            3$:     RETURN
 1057 002266                                    LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 48
SCALAR FUNCTION          ROLL

   1059 002266                              DOPINS  DEAL,100
        000040      001                             .BYTE BITSX!T.OPEN
        000040      000                             .BYTE BITSY
        000160  000000G                             .WORD   D$DEAL
   1187 002266                                      LIST
```

```
1189 002266                          MOPINS  SHAP,102
     000041     001                          .BYTE BITSX!T.OPFN
     000041     000                          .BYTE BITSY
     000162  002266'                         .WORD M$SHAP
     002266                          M$SHAP::
1190                                 .SBTTL  MIXED FUNCTION          SHAPE
1191
1192
1193 002266  013704  003120'               MOV     MONBLK,R4       ;POINT TO ARG BLOCK
1194 002272  116403  000002                MOVB    A.DIMS(R4),R3   ;GET NUMBER OF DIMS
1195 002276  010300                        MOV     R3,R0           ;COPY DIM NUMBER
1196 002300  006300                        ASL     R0              ;MULTIPLY BY 2
1197 002302  062700  000010                ADD     #A.DIM1+2,R0    ;ADD HEADER SIZE
1198 002306                                CALL    GETFRA          ;Get result, SI frame for it
1199 002312  062720  000010                ADD     #A.DIM1+2,(R0)+ ;FILL IN POINTER IN HEADER
1200 002316  012720  102001                MOV     #<A.ASB+.INTEGER>*400+1,(R0)+
1201                                                               ;FILL IN BITS
1202 002322  010320                        MOV     R3,(R0)+        ;FILL IN ALEN
1203 002324  010320                        MOV     R3,(R0)+        ;FILL IN DIM1
1204 002326  001407                        BEQ     2$              ;SORRY, NULL VECTOR
1205 002330  013701  003120'               MOV     MONBLK,R1       ;POINT TO ARG BLOCK
1206 002334  062701  000006                ADD     #A.DIM1,R1      ;POINT TO FIRST DIM
1207 002340  012120                 1$:    MOV     (R1)+,(R0)+     ;SAVE VALUE
1208 002342                                SOB     R3,1$           ;COPY ALL DIMS
1209 002346                          2$:    RETURN                  ;DONE!
1211 002350                                LIST
1212
1213
1214
1215
1216
1217 002350                          DOPINS  RHO,102
     000041     001                          .BYTE BITSX!T.OPFN
     000041     000                          .BYTE BITSY
     000162  002350'                         .WORD D$RHO
     002350                          D$RHO::
1218                                 .SBTTL  MIXED FUNCTION          RESHAPE
1219
1220
1221 002350  012700  100000                MOV     #T.VAL,R0       ;GET RESULT BITS
1222 002354  005737  003212'               TST     RARGTY          ;CHECK RIGHT TYPE
1223 002360  001402                        BEQ     1$              ;NO, LITERAL
1224 002362  052700  002000                BIS     #T.NUM,R0       ;SET NUMERIC BIT
1225 002366  010037  003142'          1$:  MOV     R0,RESBIT       ;SET RESULT BITS
1226 002372  013704  003162'               MOV     LARGBL,R4       ;POINT TO LEFT ARG BLOCK
1227 002376  005764  000004                TST     A.ALEN(R4)      ;NULL?
1228 002402  001006                        BNE     3$              ;NO, GOOD
1229 002404  012701  000001                MOV     #1,R1           ;YES, SCALARS HAVE 0 DIMENSIONS
1230 002410                                PUSH    R1              ;AND A LENGTH OF 1
1231 002412  000437                        BR      20$             ;SKIP THIS JUNK
1232
1233 002414  104411                 19$:   RANERR                  ;RANK Error
1234 002416  104417                 2$:    DOMERR                  ;YES, DOMAIN ERROR
1235 002420  013702  003170'          3$:  MOV     LARGTY,R2       ;GET LEFT ARG TYPE
1236 002424  001774                        BEQ     2$              ;OOPS!!!
1237 002426  016237  000000G 003176'       MOV     IMONCT(R2),LARGCV ;CONVERT LEFT ARG TO INTEGER
1238 002434  122764  000001  000002        CMPB    #1,A.DIMS(R4)   ;LEFT ARG A VECTOR OR SCALAR?
```

```
1239 002442  002764                        BLT     19$             ;NO, RANK ERROR
1240 002444  012701  000001                MOV     #1,R1           ;GET MULTIPLICATIVE IDENTITY
1241 002450  016403  000004                MOV     A.ALEN(R4),R3   ;GET # OF ELEMENTS
1242 002454  013702  003164'               MOV     LARGPT,R2       ;POINT TO FIRST LEFT ELEMENT
1243 002460                          4$:   CALL    @LARGCV         ;CONVERT TO INTEGER
1244 002464                                PUSH    MONVAL          ;SAVE IT
1245 002470  100752                        BMI     2$              ;LEFT ARGUMENT NEGATIVE?
1246 002472                                MUL     @SP,R1          ;MULTIPLY BY SIZE OF THIS DIMENSION
1247 002504  103525                        BCS     ROROX           ;TOO BIG!!
1248 002506                                SOB     R3,4$           ;GET SIZE OF RESULT ARRAY
1249 002512  013703  003204'          20$:  MOV     RARGBL,R3       ;POINT TO RIGHT BLOCK
1250 002516  005763  000004                TST     A.ALEN(R3)      ;NULL?
1251 002522  001002                        BNE     5$              ;NOPE, FINE
1252 002524  005701                        TST     R1              ;YES, IS RESULT A NULL?
1253 002526  001333                        BNE     2$              ;NO, CAN'T MAKE NULL BIGGER
1254
1255                                ;GET CORE FOR RESULT AND FILL IN RESULT HEADER
1256
1257 002530                          5$:   PUSH    R1              ;SAVE ARRAY LENGTH
1258 002532  013700  003212'               MOV     RARGTY,R0       ;GET RIGHT TYPE
1259 002536                                ASH     SIZTAB(R0),R1   ;TIMES ITEM SIZE
1260 002552  102502                        BVS     ROROX           ;TOO BIG
1261 002554  005201                        INC     R1              ;ROUND TO
1262 002556  042701  000001                BIC     #1,R1           ;  WORD SIZE
1263 002562  016400  000004                MOV     A.ALEN(R4),R0   ;GET DIM COUNT OF RESULT
1264 002566                                PUSH    R0              ;SAVE IT
1265 002570                                CONINC  R0              ;;CHEAT FOR SCALAR
1266 002574  006300                        ASL     R0              ;GET # OF WORDS NEEDED FOR DIMS
1267 002576  062700  000006                ADD     #A.DIM1,R0      ;ADD REST OF HEADER SIZE
1268 002602                                PUSH    R0              ;SAVE HEADER SIZE
1269 002604  060100                        ADD     R1,R0           ;ADD ON VALUE SIZE
1270 002606                                CALL    GETCOR          ;AND GET A BLOCK OF STORAGE THAT SIZE
1271 002612                                SIFRAM  ,RESBIT         ;Build an SI frame
1272 002624  010010                        MOV     R0,(R0)         ;POINT TO SELF
1273 002626  062620                        ADD     (SP)+,(R0)+     ;MAKE THAT POINTER TO DATA
1274 002630                                POP     R1              ;GET DIM COUNT
1275 002632  020127  000020                CMP     R1,#DIMAX       ;TOO MANY DIMS?
1276 002636  003267                        BGT     2$              ;YES, FLAG ERROR
1277 002640  110120                        MOVB    R1,(R0)+        ;SET DIM COUNT
1278 002642                                CONINC  R1              ;;CHEAT FOR SCALAR
1279 002646  113710  003212'               MOVB    RARGTY,(R0)     ;FILL IN BITS IN HEADER
1280 002652  152720  000200                BISB    #A.ASB,(R0)+    ;SAY IT'S AN ARRAY
1281 002656                                POP     R3              ;GET LENGTH OF RESULT
1282 002660  010320                        MOV     R3,(R0)+        ;FILL IN ALEN
1283 002662  060100                        ADD     R1,R0           ;MOVE TO LAST DIM
1284 002664  060100                        ADD     R1,R0           ;WORDS NOT BYTES!
1285 002666  012640                  9$:   MOV     (SP)+,-(R0)     ;FILL IN DIMS GOING BACKWARDS
1286 002670                                SOB     R1,9$           ;FILL IN ALL DIMS
1287
1288                                ;NOW THAT RESULT HEADER IS COMPLETE, FILL IN THE
1289                                ;ARRAY VALUES
1290
1291 002674  016501  177776                MOV     SI.VAL(SI),R1   ;POINT TO RESULT
1292 002700  016104  000004                MOV     A.ALEN(R1),R4   ;GET LENGTH
1293 002704  001424                        BEQ     16$             ;IF ANY
1294 002706  011101                        MOV     @R1,R1          ;POINT TO FIRST ELEMENT
1295 002710  013702  003204'               MOV     RARGBL,R2       ;POINT TO SOURCE ARRAY
```

```
1296 002714  016200  000004                MOV     A.ALEN(R2),R0   ;GET ITS LENGTH
1297 002720  001407                        BEQ     10$             ;IF NULL THEN SKIP
1298 002722  020004                        CMP     R0,R4           ;COPY IT ALL?
1299 002724  003401                        BLE     11$             ;YES
1300 002726  010400                        MOV     R4,R0           ;NO, ONLY COPY PART
1301 002730                        11$:    PUSH    R0              ;SAVE COUNT
1302 002732                                CALL    ACOPY           ;COPY FIRST PART
1303 002736                                POP     R0              ;GET BACK COUNT
1304 002740  160004                10$:    SUB     R0,R4           ;GET LENGTH OF REST TO COPY
1305 002742  010400                        MOV     R4,R0           ;IT IS LENGTH TO COPY NOW
1306 002744  001404                        BEQ     16$             ;IF NULL THEN DONE
1307 002746  016502  177776                MOV     SI.VAL(SI),R2   ;COPY SELF TILL DONE
1308 002752                                CALL    ACOPY           ;COPY OVER WHAT WE ALREADY HAVE
1309 002756                        16$:    RETURN
1310
1311 002760  104403                ROROX:  WSFULL                  ;NO SPACE LEFT
1313 002762                                LIST
```

```
 1315 002762                            MOPINS  RAVL,56
      000027    001                             .BYTE BITSX!T.OPFN
      000027    000                             .BYTE BITSY
      000136  002762'                           .WORD M$RAVL
      002762                            M$RAVL::
 1316                                   .SBTTL  MIXED FUNCTION          RAVEL
 1317
 1318 002762  013704  003120'                    MOV     MONBLK,R4       ;POINT TO ARG
 1319 002766                                     CALL    SAME            ;CREATE A DUPLICATE ARRAY
 1320 002772  016500  177776                     MOV     SI.VAL(SI),R0   ;GET POINTER TO IT
 1321 002776  016060  000004  000006             MOV     A.ALEN(R0),A.DIM1(R0);MAKE INTO A VECTOR
 1322 003004  112760  000001  000002             MOVB    #1,A.DIMS(R0)   ;WITH ONLY ONE DIMENSION
 1323 003012                                     RETURN
 1325 003014                                     LIST
 1326
 1327
 1328
 1329
 1330
 1331 003014                            DOPINS  CTLM,56,T.DIM
      000027    101                             .BYTE BITSX!T.OPFN
      000027    000                             .BYTE BITSY
      000136  003014'                           .WORD D$CTLM
      003014                            D$CTLM::
 1332                                   .SBTTL  MIXED FUNCTION          CATENATION/LAMINATION
 1333
 1334 003014  005037  003142'                    CLR     RESBIT          ;DIM NOT FOUND YET
 1335 003020  013702  003162'                    MOV     LARGBL,R2       ;POINT TO LEFT ARG BLOCK
 1336 003024  116201  000002                     MOVB    A.DIMS(R2),R1   ;GET # OF DIMS
 1337 003030  001003                             BNE     1$              ;It is not a scalar
 1338 003032  005201                             INC     R1              ;;CHEAT FOR SCALAR
 1339 003034  005037  003160'                    CLR     LARGFL                  ;YES, MARK AS SCALAR
 1340 003040  116200  000003             1$:     MOVB    A.BITS(R2),R0   ;GET BITS
 1341 003044  062702  000006                     ADD     #A.DIM1,R2      ;POINT TO FIRST DIM
 1342 003050  013704  003204'                    MOV     RARGBL,R4       ;POINT TO RIGHT ARG
 1343 003054  116403  000002                     MOVB    A.DIMS(R4),R3   ;GET # OF DIMS
 1344 003060  001003                             BNE     2$              ;It is not a scalar
 1345 003062  005203                             INC     R3              ;;CHEAT FOR SCALAR
 1346 003064  005037  003202'                    CLR     RARGFL          ;YES, MARK AS SCALAR
 1347 003070  116400  000003             2$:     MOVB    A.BITS(R4),R0   ;GET BITS
 1348 003074  013737  003170' 003144'            MOV     LARGTY,RESTYP   ;WHAT IS THE LEFT ARG?
 1349 003102  001004                             BNE     3$              ;IT IS NUMERIC...
 1350 003104  005737  003212'                    TST     RARGTY          ;WHAT ABOUT THE RIGHT ARG?
 1351 003110  001014                             BNE     4$              ;IT IS NUMERIC (UNFORTUNATELY)
 1352 003112  000427                             BR      CTLM0           ;...GO
 1353
 1354 003114  005737  003212'             3$:     TST     RARGTY          ;IS THE RIGHT ARG NUMERIC?
 1355 003120  001410                             BEQ     4$              ;NOPE, PROBLEM
 1356 003122  023737  003144' 003212'            CMP     RESTYP,RARGTY   ;WHAT ABOUT THE RESULT TYPE?
 1357 003130  002020                             BGE     CTLM0           ;I GUESS THE LEFT IS OK
 1358 003132  013737  003212' 003144'            MOV     RARGTY,RESTYP   ;USE THE RIGHT SIDE TYPE
 1359 003140  000414                             BR      CTLM0           ;...GO
 1360
 1361 003142  005764  000004             4$:     TST     A.ALEN(R4)      ;RIGHT ARG A NULL ARRAY?
 1362 003146  001411                             BEQ     CTLM0           ;ALSO LEGAL
 1363 003150  013737  003212' 003144'            MOV     RARGTY,RESTYP   ;ASSUME RIGHT TYPE
 1364 003156  013700  003162'                    MOV     LARGBL,R0       ;POINT TO LEFT ARG
```

```
1365 003162  005760  000004                TST     A.ALEN(R0)      ;NULL ARRAY?
1366 003166  001401                        BEQ     CTLM0           ;THAT IS LEGAL
1367
1368 003170  104417                        DOMERR                  ;NO, DOMAIN ERROR
1369 003172  062704  000006        CTLM0:  ADD     #A.DIM1,R4      ;POINT TO FIRST DIM
1370 003176  010100                        MOV     R1,R0           ;COPY # OF DIMS IN LEFT AR
1371 003200  020103                        CMP     R1,R3           ;# OF DIMS MATCH?
1372 003202  001426                        BEQ     6$              ;YES!  CONFORMITY!
1373 003204  002413                        BLT     5$              ;LEFT # OF DIMS SMALLER
1374 003206  005737  003202'              TST     RARGFL          ;RIGHT ARG A SCALAR?
1375 003212  001422                        BEQ     6$              ;YES! CONFORMITY
1376 003214  160301                        SUB     R3,R1           ;ELSE GET DIFFERENCE
1377 003216  022701  000001                CMP     #1,R1           ;N-1 DIMS?
1378 003222  001003                        BNE     4$              ;ARGS DON'T CONFORM
1379 003224  005437  003202'              NEG     RARGFL          ;ELSE SET N-1 FLAG
1380 003230  000413                        BR      6$              ;AND CONTINUE
1381
1382 003232  104412                4$:     LENERR                  ;ARGS DON'T CONFORM
1383
1384 003234  010300                5$:     MOV     R3,R0           ;MAX # OF DIMS
1385 003236  005737  003160'              TST     LARGFL          ;LEFT ARG A SCALAR?
1386 003242  001406                        BEQ     6$              ;YES, ARGS CONFORM
1387 003244  160103                        SUB     R1,R3           ;GET DIFFERENCE OF DIMS
1388 003246  022703  000001                CMP     #1,R3           ;N-1 DIMS?
1389 003252  001367                        BNE     4$              ;NOPE, LENGTH ERROR
1390 003254  005437  003160'              NEG     LARGFL          ;SET N-1 FLAG
1391
1392 003260  010037  003136'        6$:     MOV     R0,RESULT       ;SET POSSIBLE DIM ARG FLAG
1393 003264  010037  005070'              MOV     R0,REDCNT       ;SAVE MAX # OF DIMS
1394 003270  005037  005106'              CLR     OPTMP8          ;CLEAR FLAG
1395 003274  005737  003112'              TST     DIMBLK          ;DIM ARG?
1396 003300  001470                        BEQ     9$              ;NOPE
1397 003302                                PUSH    R2              ;SAVE R2-D2 FROM THE DEATH STAR
1398 003304  013700  003116'              MOV     DIMTYP,R0       ;GET THE TYPE OF THE DIMENSION
1399 003310  013702  003114'              MOV     DIMPTR,R2       ;POINT TO DATA
1400 003314                                CALL    @FMONCT(R0)     ;FETCH THE RASCAL
1401 003320                                POP     R2              ;RESTORE R2
1402 003322                                STF     AC1,AC0         ;COPY INTO REG
1403 003330                                CALL    F.TO.I          ;CONVERT
1404 003334  000407                        BR      11$             ;IT DINNA WORK ,CAPTAIN
1405 003336  013700  003154'              MOV     RESVAL,R0       ;WE HAVE THE RESULT
1406 003342  163700  000030'              SUB     ORIGIN,R0       ;SUBTRACT ORIGIN
1407 003346  005200                        INC     R0              ;CHEAT
1408 003350  003440                        BLE     7$              ;TOO SMALL
1409 003352  000443                        BR      9$              ;JUST RIGHT
1410
1411 003354                        11$:    STF     AC1,AC0         ;SCOPY IT
1412 003362                                SUBF    FORIGN,AC0      ;SUBTRACT ORIGIN
1413 003372                                ADDF    #^F1.0,AC0      ;CORRECT IT
1414 003402                                MODF    #^F1.0,AC0      ;MAKE INTO INTEGER
1415 003412                                STCFI   AC1,R0          ;FIX IT
1416 003420  100414                        BMI     7$              ;OOPS
1417 003422  103413                        BCS     7$              ;BAD DIM ARG
1418 003424  005237  005106'              INC     OPTMP8          ;INDICATE LAMINATION
1419 003430  005737  003160'              TST     LARGFL          ;LEFT ARG N-1?
1420 003434  100676                        BMI     4$              ;OOPS!
1421 003436  005737  003202'              TST     RARGFL          ;RIGHT ARG N-1?
```

```
1422 003442 100673                      BMI     4$              ;OOPS!
1423 003444 020037 005070'              CMP     R0,REDCNT       ;ARG TOO BIG?
1424 003450 003401                      BLE     8$              ;NO
1425 003452 104410              7$:     INDERR                  ;INDEX ERROR
1426 003454 005200              8$:     INC     R0              ;FUDGE LAMINATION COUNT
1427 003456 003775                      BLE     7$              ;OOPS?!
1428 003460 000403                      BR      10$             ;ALL'S OK
1429
1430 003462 020037 005070'      9$:     CMP     R0,REDCNT       ;ARG TOO BIG?
1431 003466 003371                      BGT     7$              ;YES
1432
1433 003470 012701 000001      10$:     MOV     #1,R1           ;MULTIPLICATIVE IDENTITY
1434 003474 010103                      MOV     R1,R3           ;SAVE IN R3
1435 003476 160037 003136'              SUB     R0,RESULT       ;SET DIM FLAG
1436 003502 005237 003136'              INC     RESULT          ;FUDGE FOR SOB CHECK
1437 003506 013700 005070'              MOV     REDCNT,R0       ;GET MAX # OF DIMS
1438
1439
1440 003512 020037 003136'      CTLMLP: CMP     R0,RESULT       ;THIS THE MAGIC DIM?
1441 003516 001046                      BNE     4$              ;NOPE
1442 003520 005737 003160'              TST     LARGFL          ;LEFT ARG OK?
1443 003524 003405                      BLE     1$              ;NO, COUNT STARTS AT 1
1444 003526                             MUL     (R2)+,R1        ;ELSE COUNT STARTS WITH LEFT DIM
1445 003540 005737 003202'      1$:     TST     RARGFL          ;RIGHT ARG OK?
1446 003544 003405                      BLE     2$              ;NO
1447 003546                             MUL     (R4)+,R3        ;INITIALIZE RIGHT COUNT
1448 003560 005737 005106'      2$:     TST     OPTMP8          ;IS THIS LAMINATION?
1449 003564 001005                      BNE     3$              ;IS LAMINATION
1450 003566 010146                      MOV     R1,-(SP)        ;CATENATION, PUSH LEFT COUNT
1451 003570 060316                      ADD     R3,@SP          ;ADD RIGHT COUNT
1452 003572 010537 003142'              MOV     SI,RESBIT       ;SET FOUND FLAG
1453 003576 000767                      BR      8$              ;AND CYCLE
1454 003600 012746 000002      3$:      MOV     #2,-(SP)        ;PUSH NEW LAMINATION DIM
1455 003604 005237 005070'              INC     REDCNT          ;1 MORE DIM IN RESULT
1456 003610 010537 003142'              MOV     SI,RESBIT       ;SET FOUND FLAG
1457 003614 005737 003160'              TST     LARGFL          ;LEFT ARG A SCALAR?
1458 003620 001401                      BEQ     19$             ;NO
1459 003622 005742                      TST     -(R2)           ;CORRECT POINTER
1460 003624 005737 003202'      19$:    TST     RARGFL          ;RIGHT ARG A SCALAR?
1461 003630 001401                      BEQ     4$              ;YES
1462 003632 005744                      TST     -(R4)           ;CORRECT RIGHT POINTER
1463
1464 003634 005737 003160'      4$:     TST     LARGFL          ;LEFT ARG A SCALAR?
1465 003640 001012                      BNE     5$              ;NO
1466 003642 011446                      MOV     @R4,-(SP)       ;PUSH RIGHT DIM
1467 003644 020037 003136'              CMP     R0,RESULT       ;COUNTING?
1468 003650 003041                      BGT     7$              ;NOPE
1469 003652                             MUL     @R4,R1          ;MULTIPLY LEFT BY RIGHT COUNT
1470 003664 000423                      BR      6$              ;BUMP RIGHT COUNT
1471
1472 003666 011246              5$:     MOV     @R2,-(SP)       ;PUSH LEFT DIM
1473 003670 020037 003136'              CMP     R0,RESULT       ;COUNTING?
1474 003674 003027                      BGT     7$              ;NO
1475 003676 001405                      BEQ     9$              ;SPECIAL LAMINATION CASE?
1476 003700                             MUL     @R2,R1          ;BUMP LEFT COUNT
1477 003712 005737 003202'      9$:     TST     RARGFL          ;RIGHT ARG A SCALAR?
1478 003716 001006                      BNE     6$              ;NO
```

```
1479 003720                                    MUL     @R2,R3          ;YES, BUMP RIGHT COUNT BY LEFT DIM
1480 003732  000410                            BR      7$              ;AND BUMP POINTERS
1481
1482 003734  020037  003136'          6$:     CMP     R0,RESULT       ;IS THIS SPECIAL LAMINATION?
1483 003740  001405                            BEQ     7$              ;YES
1484 003742                                    MUL     @R4,R3          ;RIGHT COUNT * RIGHT DIM
1485
1486 003754  022224                    7$:     CMP     (R2)+,(R4)+     ;DIMENSIONS MATCH?
1487 003756  001407                            BEQ     8$              ;YES, FINE
1488 003760  005737  003160'                    TST     LARGFL          ;LEFT ARG OK?
1489 003764  001404                            BEQ     8$              ;SCALAR - GOES W/ ANYTHING
1490 003766  005737  003202'                    TST     RARGFL          ;RIGHT ARG OK?
1491 003772  001401                            BEQ     8$              ;OK HERE
1492
1493 003774  104412                  15$:     LENERR                  ;LENGTH ERROR
1494
1495 003776  005300                    8$:     DEC     R0              ;DONE YET?
1496 004000  003402                            BLE     17$             ;YES
1497 004002  000137  003512'                    JMP     CTLMLP          ;NOPE
1498 004006  005737  003142'          17$:     TST     RESBIT          ;FOUND FLAG SET?
1499 004012  001004                            BNE     13$             ;YES
1500 004014  012746  000002                    MOV     #2,-(SP)        ;NO, PUSH NEW LAMINATION DIM
1501 004020  005237  005070'                    INC     REDCNT          ;INCREASE # OF RESULT DIMS
1502 004024  013702  003164'          13$:     MOV     LARGPT,R2       ;POINT TO FIRST ELEMENT
1503 004030  013704  003206'                    MOV     RARGPT,R4       ;POINT TO FIRST ELEMENT
1504 004034  010137  003302'          CTLM1:   MOV     R1,NINPT1       ;SAVE LEFT COUNT
1505 004040  010337  003300'                    MOV     R3,NINPT2       ;SAVE RIGHT COUNT
1506 004044  013700  005070'                    MOV     REDCNT,R0       ;GET # OF RESULT DIMS
1507 004050  012701  000001                    MOV     #1,R1           ;GET MULTIPLICATIVE IDENTITY
1508 004054  010603                            MOV     SP,R3           ;COPY STACK PTR
1509 004056                            1$:     MUL     (R3)+,R1        ;MUL BY NEXT DIM
1510 004070                                    SOB     R0,1$           ;CALCULATE ALEN
1511 004074                                    PUSH    R1              ;SAVE RESULT ALEN
1512 004076  010137  003142'                    MOV     R1,RESBIT       ;SAVE ALEN
1513 004102  013700  003144'                    MOV     RESTYP,R0       ;GET RESULT TYPE
1514 004106                                    ASH     SIZTAB(R0),R1   ;CALCULATE # OF BYTES
1515 004122  102004                            BVC     3$              ;NOT TOO BIG
1516 004124  104412                            LENERR                  ;TOO BIG!
1517 004126  005201                    2$:     INC     R1              ;ROUND UP BYTE COUNT
1518 004130  042701  000001                    BIC     #1,R1           ;CLEAR FOR WORD BOUNDARY
1519 004134  013700  005070'            3$:     MOV     REDCNT,R0       ;GET # OF DIMS
1520 004140  006300                            ASL     R0              ;CONVERT FOR WORDS
1521 004142  062700  000006                    ADD     #A.DIM1,R0      ;ADD HEADER OVERHEAD
1522 004146                                    PUSH    R0              ;SAVE OFFSET
1523 004150  060100                            ADD     R1,R0           ;GET TOTAL SIZE
1524 004152                                    CALL    GETFRA          ;Get result, SI frame for it
1525 004156  062620                            ADD     (SP)+,(R0)+     ;ADD IN OFFSET
1526 004160  113720  005070'                    MOVB    REDCNT,(R0)+    ;FILL IN # OF DIMS
1527 004164  113710  003144'                    MOVB    RESTYP,(R0)     ;BITS
1528 004170  001003                            BNE     4$              ;YES, FINE
1529 004172  042765  002000  177774            BIC     #T.NUM,SI.BIT(SI);RESULT NOT NUMERIC
1530 004200  152720  000200            4$:     BISB    #A.ASB,(R0)+    ;MARK AS ARRAY
1531 004204  013703  003204'                    MOV     RARGBL,R3       ;POINT TO RIGHT BLOCK
1532 004210  005763  000004                    TST     A.ALEN(R3)      ;EMPTY ARRAY?
1533 004214  001002                            BNE     6$              ;NO, FINE
1534 004216  013703  003162'                    MOV     LARGBL,R3       ;YES, POINT TO LEFT ARG
1535 004222  012620                    6$:     MOV     (SP)+,(R0)+     ;FILL IN ALEN
```

```
1536 004224  017500  177776                    MOV     @SI.VAL(SI),R0  ;POINT PAST LAST DIM
1537 004230  013703  005070'                    MOV     REDCNT,R3       ;GET DIM COUNT
1538 004234  012640                    5$:     MOV     (SP)+,-(R0)     ;FILL IN DIMS
1539 004236                                    SOB     R3,5$           ;BACK TO FRONT
1540 004242  013703  003302'                    MOV     NINPT1,R3       ;GET LEFT COUNT
1541 004246  063703  003300'                    ADD     NINPT2,R3       ;ADD RIGHT COUNT
1542 004252  005000                            CLR     R0              ;CLEAR FOR DIVIDE
1543 004254  013701  003142'                    MOV     RESBIT,R1       ;GET ALEN
1544 004260                                    DIV     R3,R0           ;DIVIDE BY COUNT SUM
1545 004272  010037  003136'                    MOV     R0,RESULT       ;SAVE # OF CYCLES
1546 004276  001510                            BEQ     CTLMX           ;CATENATION OF 2 NULLS!!!!!
1547                                           .ENABL  LSB
1548                                   ;
1549                                   ;DO THE ACTUAL CATENATION.  COPY SOME FROM THE LEFT,
1550                                   ;SOME FROM THE RIGHT, SOME FROM THE LEFT.....
1551                                   ;
1552 004300  017501  177776            CTLMN:  MOV     @SI.VAL(SI),R1  ;POINT TO RESULT
1553 004304  013700  003144'                    MOV     RESTYP,R0       ;GET RESULT TYPE
1554 004310  006300                            ASL     R0              ;TIMES 2
1555 004312  006300                            ASL     R0              ;     "   4
1556 004314  013702  003170'                    MOV     LARGTY,R2       ;GET LEFT ARG TYPE
1557 004320  013703  003212'                    MOV     RARGTY,R3       ;GET RIGHT ARG TYPE
1558 004324  060002                            ADD     R0,R2           ;ADD RESULT TO LEFT ARG FOR TABLE LOOKUP
1559 004326  060003                            ADD     R0,R3           ;SAME FOR RIGHT ARG
1560 004330  063700  003144'                    ADD     RESTYP,R0       ;RESULT STORE ROUTINE
1561 004334  016237  000000G 003176'           MOV     LARGCT(R2),LARGCV ;LEFT FETCH ROUTINE
1562 004342  016337  000000G 003220'           MOV     RARGCT(R3),RARGCV ;RIGHT FETCH ROUTINE
1563 004350  016037  000000G 003152'           MOV     RESCT(R0),RESCVT;AH, HERE IT IS
1564 004356  013702  003164'                    MOV     LARGPT,R2       ;POINT TO LEFT ARG
1565 004362  013703  003206'                    MOV     RARGPT,R3       ;POINT TO RIGHT ARG
1566 004366  005737  003160'                    TST     LARGFL          ;MAKE POSITIVE VALUE
1567 004372  002002                            BGE     1$              ;...NON-NEGATIVE
1568 004374  005437  003160'                    NEG     LARGFL          ;IT WAS NEGATIVE
1569 004400  005737  003202'            1$:     TST     RARGFL          ;MAKE POSITIVE VALUE
1570 004404  002002                            BGE     2$              ;...NON-NEGATIVE
1571 004406  005437  003202'                    NEG     RARGFL          ;IT WAS NEGATIVE
1572
1573 004412  013700  003302'            2$:     MOV     NINPT1,R0       ;COUNT FOR LOOP
1574 004416  001415                            BEQ     4$              ;...NONE...
1575 004420                              3$:     CALL    @LARGCV         ;GET LEFT ARG
1576 004424  013737  003200' 003154'            MOV     LARGVA,RESVAL   ;IF INTEGER OR BOOLEAN
1577 004432                                    STF     AC1,AC0         ;IF FLOATING POINT
1578 004440                                    CALL    @RESCVT         ;SAVE IT
1579 004444  104402                            SYSERR                  ;REALLY, TRULY, SHOULDN'T HAPPEN
1580 004446                                    SOB     R0,3$           ;LOOP
1581
1582 004452  013700  003300'            4$:     MOV     NINPT2,R0       ;COUNT FOR LOOP
1583 004456  001415                            BEQ     6$              ;...NONE...
1584 004460                              5$:     CALL    @RARGCV         ;GET RIGHT ARG
1585 004464  013737  003222' 003154'            MOV     RARGVA,RESVAL   ;IF INTEGER OR BOOLEAN
1586 004472                                    STF     AC2,AC0         ;IF FLOATING POINT
1587 004500                                    CALL    @RESCVT         ;SAVE IT
1588 004504  104402                            SYSERR                  ;THIS ALSO A NO-NO
1589 004506                                    SOB     R0,5$           ;LOOP
1590 004512  005337  003136'            6$:     DEC     RESULT          ;DECREMENT COUNT
1591 004516  003335                            BGT     2$              ;AND CONTINUE
1592                                           .DSABL  LSB
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 50-5
MIXED FUNCTION          CATENATION/LAMINATION

    1593
    1594 004520                        CTLMX:  RETURN                  ;ALL DONE
    1596 004522                                LIST
```

```
1598 004522                            MOPINS  IOTA,72
     000035     001                             .BYTE BITSX!T.OPFN
     000035     000                             .BYTE BITSY
     000152  004522'                            .WORD M$IOTA
     004522                            M$IOTA::
1599                                   .SBTTL  MIXED FUNCTION          IOTA -- INDEX GENERATOR
1600
1601
1602 004522  013704  003120'                    MOV     MONBLK,R4       ;GET ARG BLOCK POINTER
1603 004526  022764  000001  000004             CMP     #1,A.ALEN(R4)   ;SCALAR?
1604 004534  001401                            BEQ     2$              ;YES, FINE
1605 004536  104417                    1$:     DOMERR                  ;DOMAIN ERROR
1606 004540  011402                    2$:     MOV     (R4),R2         ;POINT TO ARG
1607 004542  013704  003124'                    MOV     MONTYP,R4       ;GET THE SOURCE TYPE
1608 004546  001773                            BEQ     1$              ;CHARACTER? I DON'T UNDERSTAND...
1609 004550                                    CALL    @IMONCT(R4)     ;CONVERT SCALAR TO INTEGER
1610 004554  013703  003134'                    MOV     MONVAL,R3       ;GET NUMBER
1611 004560  100766                            BMI     1$              ;BAD ARG
1612 004562  010300                            MOV     R3,R0           ;COPY ARG
1613 004564  006300                            ASL     R0              ;MULTIPLY BY 2
1614 004566  102425                            BVS     5$              ;Too Big
1615 004570  062700  000010                    ADD     #A.DIM1+2,R0    ;ADD HEADER SIZE
1616 004574  102422                            BVS     5$              ;TOO BIG
1617 004576                                    CALL    GETFRA          ;Build result, SI frame for it
1618 004602  062720  000010                    ADD     #A.DIM1+2,(R0)+ ;SET POINTER TO FIRST VALUE
1619 004606  112720  000001                    MOVB    #1,(R0)+        ;ONE DIMENSION (VECTOR)
1620 004612  112720  000204                    MOVB    #.INTEGER+A.ASB,(R0)+ ;ARRAY IS INTEGER
1621 004616  010320                            MOV     R3,(R0)+        ;VECTOR LENGTH
1622 004620  010320                            MOV     R3,(R0)+        ;LENGTH OF FIRST DIMENSION
1623 004622  001406                            BEQ     4$              ;IF NO LEN, THEN DONE
1624 004624  013702  000030'                    MOV     ORIGIN,R2       ;STARTING POINT
1625 004630  010220                    3$:     MOV     R2,(R0)+        ;STORE NEXT VALUE
1626 004632  005202                            INC     R2              ;INCREMENT VALUE
1627 004634                                    SOB     R3,3$           ;AND FILL IN VECTOR
1628 004640                            4$:     RETURN                  ;ALL DONE
1629
1630 004642  104403                    5$:     WSFULL                  ;ACTUAL MESSAGE
1632 004644                                    LIST
1633
1634
1635
1636
1637
1638 004644                            DOPINS  IND,72
     000035     001                             .BYTE BITSX!T.OPFN
     000035     000                             .BYTE BITSY
     000152  000000G                            .WORD   D$IND
1699 004644                                    LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 52
MIXED FUNCTION           IOTA -- INDEX GENERATOR

   1701 004644                              DOPINS  1COM,124
        000052      001                             .BYTE BITSX!T.OPFN
        000052      000                             .BYTE BITSY
        000204  000000G                             .WORD   D$1COM
   1709 004644                                      LIST
   1710
   1711
   1712
   1713
   1714
   1715 004644                              DOPINS  COMP,60,T.DIM
        000030      101                             .BYTE BITSX!T.OPFN
        000030      000                             .BYTE BITSY
        000140  000000G                             .WORD   D$COMP
   1789 004644                                      LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 53
MIXED FUNCTION          IOTA -- INDEX GENERATOR

   1791 004644                              DOPINS  1EXP,130
        000054      001                             .BYTE BITSX!T.OPFN
        000054      000                             .BYTE BITSY
        000210  000000G                             .WORD   D$1EXP
   1799 004644                                      LIST
   1800
   1801
   1802
   1803
   1804
   1805 004644                              DOPINS  EXPD,62,T.DIM
        000031      101                             .BYTE BITSX!T.OPFN
        000031      000                             .BYTE BITSY
        000142  000000G                             .WORD   D$EXPD
   1864 004644                                      LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 54
MIXED FUNCTION          IOTA -- INDEX GENERATOR

   1866 004644                              DOPINS  DROP,104
        000042      001                             .BYTE BITSX!T.OPFN
        000042      000                             .BYTE BITSY
        000164  000000G                             .WORD   D$DROP
   1907 004644                                      LIST
   1908
   1909
   1910
   1911 004644                              DOPINS  TAKE,114
        000046      001                             .BYTE BITSX!T.OPFN
        000046      000                             .BYTE BITSY
        000174  000000G                             .WORD   D$TAKE
   2149 004644                                      LIST
```

```
   2151 004644                          MOPINS  1REV,146
        000063     001                          .BYTE BITSX!T.OPFN
        000063     000                          .BYTE BITSY
        000226  000000G                              .WORD   M$1REV
   2160 004644                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 004644                          MOPINS  REVS,144,T.DIM
        000062     101                          .BYTE BITSX!T.OPFN
        000062     000                          .BYTE BITSY
        000224  000000G                              .WORD   M$REVS
   2204 004644                                  LIST
```

```
   2206 004644                          DOPINS  1ROT,146
        000063     001                          .BYTE BITSX!T.OPFN
        000063     000                          .BYTE BITSY
        000226  000000G                         .WORD   D$1ROT
   2214 004644                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 004644                          DOPINS  ROT,144,T.DIM
        000062     101                          .BYTE BITSX!T.OPFN
        000062     000                          .BYTE BITSY
        000224  000000G                         .WORD   D$ROT
   2328 004644                                  LIST
```

```
   2330 004644                              MOPINS  TRAN,126
        000053     001                              .BYTE BITSX!T.OPFN
        000053     000                              .BYTE BITSY
        000206  000000G                                  .WORD   M$TRAN
   2415 004644                                      LIST
```

```
   2417 004644                              DOPINS  TRAN,126
        000053     001                              .BYTE BITSX!T.OPFN
        000053     000                              .BYTE BITSY
        000206  000000G                             .WORD   D$TRAN
   2654 004644                                      LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 59
MIXED FUNCTION           IOTA -- INDEX GENERATOR

   2656 004644                              DOPINS  ENCD,76
        000037      001                             .BYTE BITSX!T.OPFN
        000037      000                             .BYTE BITSY
        000156  000000G                             .WORD   D$ENCD
   2856 004644                                      LIST
```

```
 2858 004644                            DOPINS  DECD,64
      000032     001                             .BYTE BITSX!T.OPFN
      000032     000                             .BYTE BITSY
      000144  000000G                            .WORD   D$DECD
 2988 004644                                     LIST
```

```
 2990 004644                            MOPINS  GDWN,136,T.DIM
      000057     101                             .BYTE BITSX!T.OPFN
      000057     000                             .BYTE BITSY
      000216  000000G                                .WORD   M$GDWN
 3001 004644                                     LIST
 3002
 3003
 3004 004644                            MOPINS  GDUP,140,T.DIM
      000060     101                             .BYTE BITSX!T.OPFN
      000060     000                             .BYTE BITSY
      000220  000000G                                .WORD   M$GDUP
 3219 004644                                     LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 62
MIXED FUNCTION          IOTA -- INDEX GENERATOR

   3221 004644                          MOPINS  MMD,122
        000051      001                         .BYTE BITSX!T.OPFN
        000051      000                         .BYTE BITSY
        000202  000000G                              .WORD   M$MMD
   3316 004644                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 63
MIXED FUNCTION          IOTA -- INDEX GENERATOR

   3318 004644                              DOPINS  DMD,122
        000051      001                             .BYTE BITSX!T.OPFN
        000051      000                             .BYTE BITSY
        000202  000000G                             .WORD   D$DMD
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 64
MIXED FUNCTION          IOTA -- INDEX GENERATOR

   3864 004644                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 65
MIXED FUNCTION          IOTA -- INDEX GENERATOR

  3866 004644                            DOPINS  MEM,70
       000034      001                           .BYTE BITSX!T.OPFN
       000034      000                           .BYTE BITSY
       000150  000000G                           .WORD   D$MEM
  3921 004644                                    LIST
```

```
 3923 004644                            MOPINS  NUB,106
      000043     001                             .BYTE BITSX!T.OPFN
      000043     000                             .BYTE BITSY
      000166  000000G                                 .WORD   M$NUB
 3986 004644                                     LIST
 3987
 3988
 3989
 3990
 3991
 3992 004644                            DOPINS  UNON,106
      000043     001                             .BYTE BITSX!T.OPFN
      000043     000                             .BYTE BITSY
      000166  000000G                            .WORD   D$UNON
 4022 004644                                     LIST
 4023
 4024
 4025
 4026
 4027
 4028 004644                            DOPINS  INTR,66
      000033     001                             .BYTE BITSX!T.OPFN
      000033     000                             .BYTE BITSY
      000146  000000G                            .WORD   D$INTR
 4042 004644                                     LIST
 4043
 4044
 4045
 4046
 4047
 4048 004644                            DOPINS  SDIF,50
      000024     001                             .BYTE BITSX!T.OPFN
      000024     000                             .BYTE BITSY
      000130  000000G                            .WORD   D$SDIF
 4161 004644                                     LIST
```

```
4163 004644                             DOPINS  SCON,234
     000116     001                             .BYTE BITSX!T.OPFN
     000116     000                             .BYTE BITSY
     000314  000000G                             .WORD   D$SCON
4179 004644                                     LIST
4180
4181
4182 004644                             DOPINS  CONT,112
     000045     001                             .BYTE BITSX!T.OPFN
     000045     000                             .BYTE BITSY
     000172  000000G                             .WORD   D$CONT
4200 004644                                     LIST
4201
4202
4203 004644                             DOPINS  PSUB,236
     000117     001                             .BYTE BITSX!T.OPFN
     000117     000                             .BYTE BITSY
     000316  000000G                             .WORD   D$PSUB
4209 004644                                     LIST
4210
4211
4212 004644                             DOPINS  SUBS,116
     000047     001                             .BYTE BITSX!T.OPFN
     000047     000                             .BYTE BITSY
     000176  000000G                             .WORD   D$SUBS
4288 004644                                     LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 68
MIXED FUNCTION           IOTA -- INDEX GENERATOR

 4290 004644                                MOPINS  EXE,70
      000034      001                               .BYTE BITSX!T.OPFN
      000034      000                               .BYTE BITSY
      000150  004644'                               .WORD M$EXE
      004644                                M$EXE::
 4291                                       .SBTTL  MIXED FUNCTION          EXECUTE
 4292
 4293                                       ;
 4294                                       ;THE EXECUTE OPERATOR IS USED TO EVALUATE A STRING.  THE RESULT
 4295                                       ;IS THE SAME AS IF THE USER HAD TYPED THE LINE.  AN EXECUTION
 4296                                       ;FRAME IS SPAWNED ON THE SI STACK, AND THE INPUT STREAM (INPTR)
 4297                                       ;IS SWITCHED TO E$CHR WHICH CAUSES THE STRING TO BE EVALUATED.
 4298                                       ;
 4299                                       ;
 4300
 4301 004644  016500  177776                       MOV     SI.VAL(SI),R0   ;POINT TO ARG
 4302 004650  132760  000006  000003               BITB    #A.NUM,A.BITS(R0);NUMERIC?
 4303 004656  001404                               BEQ     1$              ;NO
 4304 004660  010004                               MOV     R0,R4           ;YES, SO RESULT IS JUST ARG
 4305 004662                                       CALL    SAME            ;RESULT IS SAME AS ARG
 4306 004666                                       RETURN                  ;ALL DONE
 4307
 4308 004670  023705  006346'              1$:     CMP     SITOP,SI        ;STACK OVERFLOW?
 4309 004674  101002                               BHI     2$              ;NO
 4310 004676                                       CALL    STKXTN          ;EXTEND THE STACK
 4311 004702  013700  003314'              2$:     MOV     SIFRAM,R0       ;POINT TO CURRENT FRAME
 4312 004706  013760  004764'  000002             MOV     FLAGS,F.FLAG(R0);SAVE FRAME'S FLAGS
 4313 004714  012737  000400  004764'             MOV     #F$EXEC,FLAGS   ;THIS IS AN EXECUTION FRAME
 4314 004722  012625                               MOV     (SP)+,(SI)+     ;RETURN GOES INTO PARSE FRAME
 4315 004724  005025                               CLR     (SI)+           ;WE HAVE NO VALUE YET
 4316 004726  005025                               CLR     (SI)+           ;BUT WE WILL HAVE LATER
 4317 004730  010537  003314'                      MOV     SI,SIFRAM       ;POINT TO REF. PT. OF EXECUTION FRAME
 4318 004734  010501                               MOV     SI,R1           ;COPY PTR TO REF. PT.
 4319 004736  010025                               MOV     R0,(SI)+        ;POINT BACK TO PREVIOUS FRAME
 4320 004740  005025                               CLR     (SI)+           ;NO FLAGS
 4321 004742  005025                               CLR     (SI)+           ;LAST C-BYTE
 4322 004744  005025                               CLR     (SI)+           ;HEAD POINTER
 4323 004746                                       POP     <(SI)+,(SI)+,(SI)+,(SI)+>
 4324                                                                       ;Pop R2, R3, R4, RETURN
 4325 004756  013725  000064'                      MOV     INPTR,(SI)+     ;OLD INPUT STREAM POINTER
 4326 004762  013725  005060'                      MOV     E$PT,(SI)+      ;OLD INPUT POINTER INTO STRING
 4327 004766  013725  005056'                      MOV     E$CT,(SI)+      ;OLD COUNT OF INPUT CHARS
 4328 004772  010561  000006                       MOV     SI,F.HEAD(R1)   ;LENGTH OF HEADER
 4329
 4330 004776  012737  000000G 000064'             MOV     #E$CHR,INPTR    ;POINT TO NEW INPUT ROUTINE
 4331 005004  013700  003120'                      MOV     MONBLK,R0       ;POINT TO ARG
 4332 005010  016037  000004  005056'             MOV     A.ALEN(R0),E$CT ;LENGTH OF INPUT
 4333 005016  011037  005060'                      MOV     @R0,E$PT        ;POINT TO FIRST CHAR OF INPUT
 4334 005022  000137  000000G                      JMP     EXEC            ;''AND AWAAAAY WE GO...''  (J.G.)
 4336 005026                                       LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 69
MIXED FUNCTION           EXECUTE

   4338 005026                              MOPINS  ENCD,76
        000037     001                              .BYTE BITSX!T.OPFN
        000037     000                              .BYTE BITSY
        000156  000000G                                  .WORD   M$ENCD
   4411 005026                              LIST
```

```
 4413 005026                            MOPINS  FMT,160
      000070     001                             .BYTE BITSX!T.OPFN
      000070     000                             .BYTE BITSY
      000240  000000G                                .WORD   M$FMT
 4423 005026                                     LIST
 4424
 4425
 4426
 4427
 4428
 4429 005026                            DOPINS  FMT,160
      000070     001                             .BYTE BITSX!T.OPFN
      000070     000                             .BYTE BITSY
      000240  000000G                            .WORD   D$FMT
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 71
MIXED FUNCTION          EXECUTE

    4953 005026                                 LIST
```

```
   4955 005026                          MOPINS  IBM,132,,,MONIBM
        000055      001                          .BYTE BITSX!T.OPFN
        000055      000                          .BYTE BITSY
        000212  000000G                              .WORD   MONIBM
   4958 005026                                  LIST
   4959
   4960
   4961 005026                          DOPINS  IBM,132
        000055      001                          .BYTE BITSX!T.OPFN
        000055      000                          .BYTE BITSY
        000212  000000G                          .WORD   D$IBM
   5046 005026                                  LIST
```

```
005026                          MOPINS  QUAD,74,T.NIL+T.SPOP
000036    045                           .BYTE BITSX!T.OPFN
000036    000                           .BYTE BITSY
000154  005026'                         .WORD M$QUAD
005026                          M$QUAD::
                                .SBTTL  MIXED FUNCTION          QUAD INPUT

005026  023705  006346'               CMP     SITOP,SI        ;STACK OVERFLOW?
005032  101002                        BHI     1$              ;NO
005034                                CALL    STKXTN          ;YES, EXTEND THE STACK
005040  013700  003314'          1$:  MOV     SIFRAM,R0       ;POINT TO CURRENT FRAME
005044  013760  004764'  000002       MOV     FLAGS,F.FLAG(R0);SAVE ITS FLAGS
005052  012625                        MOV     (SP)+,(SI)+     ;SET RETURN
005054  005025                        CLR     (SI)+           ;NO BITS YET
005056  005025                        CLR     (SI)+           ;NO VALUE YET
005060  010537  003314'               MOV     SI,SIFRAM       ;POINT TO NEW EVAL FRAME
005064  010025                        MOV     R0,(SI)+        ;POINT TO PREVIOUS FRAME
005066  005025                        CLR     (SI)+           ;CLEAR FLAGS
005070  005025                        CLR     (SI)+           ;CLEAR LAST C-BYTE
005072  005025                        CLR     (SI)+           ;HEADER POINTER
005074                                POP     <(SI)+,(SI)+,(SI)+,(SI)+>
                                                            ;Push R2,R3,R4,RETURN
005104  013725  000064'               MOV     INPTR,(SI)+     ;PUSH OLD INPUT POINTER
005110  013725  005060'               MOV     E$PT,(SI)+      ;EXECUTE INPUT STRING POINTER
005114  013725  005056'               MOV     E$CT,(SI)+      ;AND COUNT (SO WE CAN EXECUTE QUAD INPUT!)
005120  012737  000000G 000064'       MOV     #GETL,INPTR     ;QUAD INPUT ALWAYS COMES FROM KEYBOARD!
005126  012737  000200  004764'       MOV     #F$EVI,FLAGS    ;MARK FRAME TYPE
005134  013700  003314'               MOV     SIFRAM,R0       ;POINT TO REF POINT OF FRAME
005140  010560  000006                MOV     SI,F.HEAD(R0)   ;POINT TO WHERE CODESTRING GOES
005144  000137  000000G               JMP     EXEC            ;AND EVALUATE SOME INPUT
005150                                LIST


005150                          DOPINS  QUAD,74,T.NIL+T.SPOP
000036    045                           .BYTE BITSX!T.OPFN
000036    000                           .BYTE BITSY
000154  005150'                         .WORD D$QUAD
005150                          D$QUAD::
                                .SBTTL  MIXED FUNCTION          QUAD OUTPUT

005150  042737  001000  000000'       BIC     #SY.PMT,SYSTAT  ;No Special Prompt
005156                                PUSH    OCHAN           ;Save Output Channel
005162  012737  000256' 005052'       MOV     #CHAN00,OCHAN   ;Point to Terminal
005170  023737  000246' 000250'       CMP     CHAN00+C.WIDC,CHAN00+C.WIDR
                                                              ;Beginning of line?
005176  001402                        BEQ     1$              ;Yes, no problem
005200                                CALL    NEWLIN          ;Output a <CR><LF>
005204  010500                  1$:   MOV     SI,R0           ;POINT TO ARRAY TO PRINT
005206                                CALL    PRINT           ;PRINT EXPRESSION
005212                                CALL    NEWLIN          ;    AND END THE LINE
005216                                POP     OCHAN           ;Restore Output Channel
005222                                RETURN                  ;Finished
005224                                LIST
```

```
                                        MOPINS  QQAD,142,T.NIL+T.SPOP
000061     045                                  .BYTE BITSX!T.OPFN
000061     000                                  .BYTE BITSY
000222  005224'                                 .WORD M$QQAD
005224                                  M$QQAD::
                                        .SBTTL  MIXED FUNCTION          QUOTE QUAD INPUT


005224                                          PUSH    <INPTR,ICHAN>   ;Save Current Input Pointer, Channel
005234  012737  000000G 000064'                 MOV     #GETL,INPTR     ;INPUT ALWAYS COMES FROM KEYBOARD!
005242  012737  000256' 005050'                 MOV     #CHAN00,ICHAN   ;Get Input from the Console
005250                                          CALL    GETLIN          ;GET A LINE OF INPUT
005254                                          POP     <ICHAN,INPTR>   ;Restore Old Input Channel, Pointer
005264                                          CALL    FIXLIN          ;FIX IT UP
005270  122744  000015                         CMPB    #CHR.CR,-(R4)   ;LAST CHAR A CR?
005274  001001                                  BNE     1$              ;NO
005276  005301                                  DEC     R1              ;YES, DELETE IT
005300                                  1$:     PUSH    <R0,R1>         ;PUSH PTR AND COUNT
005304  012700  000010                         MOV     #A.DIM1+2,R0    ;GET HEADER SIZE
005310                                          PUSH    R0              ;SAVE FOR OFFSET
005312  005201                                  INC     R1              ;BUMP COUNT
005314  042701  000001                         BIC     #1,R1           ;TO ROUND FOR WORD BOUNDARY
005320  060100                                  ADD     R1,R0           ;GET TOTAL ROOM NEEDED
005322                                          CALL    GETFRA          ;GET CORE FOR RESULT
005326  042765  002000  177774                 BIC     #T.NUM,SI.BITS(SI)
                                                                        ;Not Numeric
005334  062620                                  ADD     (SP)+,(R0)+     ;Add In Offset
005336  012720  100001                         MOV     #A.ASB*400+1,(R0)+ ;FILL IN # OF DIMS/MARK AS LITERAL
005342                                          POP     <R1,R2>         ;GET ALEN, POINTER
005346  010120                                  MOV     R1,(R0)+        ;FILL IN ALEN
005350  010120                                  MOV     R1,(R0)+        ;FILL IN DIM1
005352  001403                                  BEQ     3$              ;NULL!?
005354  112220                          2$:     MOVB    (R2)+,(R0)+     ;COPY 1 CHAR
005356                                          SOB     R1,2$           ;COPY ENTIRE LINE
005362                                  3$:     RETURN                  ;ALL DONE
005364                                          LIST


                                        DOPINS  QQAD,142,T.NIL+T.SPOP
000061     045                                  .BYTE BITSX!T.OPFN
000061     000                                  .BYTE BITSY
000222  005364'                                 .WORD D$QQAD
005364                                  D$QQAD::
                                        .SBTTL  MIXED FUNCTION          QUOTE QUAD OUTPUT

005364  052737  001000  000000'                 BIS     #SY.PMT,SYSTAT  ;Indicate Prompt
005372                                          PUSH    OCHAN           ;Save Output Channel
      76  012737  000256' 005052'                 MOV     #CHAN00,OCHAN   ;Point to Terminal
005404  010500                                  MOV     SI,R0           ;POINT TO ARRAY TO PRINT
005406                                          CALL    PRINT           ;PRINT EXPRESSION
```

```
 5149 005412                                    POP     OCHAN           ;Restore Output Channel
 5150 005416                                    RETURN                  ;Finished
 5152 005420                                    LIST
 5153
 5154
 5155
 5156
 5157
 5158 005420                            MOPINS  DQAD,134,T.NIL+T.SPOP
      000056     045                            .BYTE BITSX!T.OPFN
      000056     000                            .BYTE BITSY
      000214  000000G                               .WORD   M$DQAD
 5192 005420                                    LIST
 5193
 5194
 5195
 5196 005420                            DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
      000056     045                            .BYTE BITSX!T.OPFN
      000056     000                            .BYTE BITSY
      000214  005364'                           .WORD   D$QQAD
 5199 005420                                    LIST
```

```
 5201 005420                            MOPINS  BXAV,162,T.NIL
      000071      005                           .BYTE BITSX!T.OPFN
      000071      000                           .BYTE BITSY
      000242  000000G                               .WORD   M$BXAV
 5219 005420                                    LIST
 5220
 5221
 5222
 5223
 5224
 5225 005420                            MOPINS  BXCT,164,T.NIL+T.SPOP
      000072      045                           .BYTE BITSX!T.OPFN
      000072      000                           .BYTE BITSY
      000244  000000G                               .WORD   M$BXCT
 5234 005420                                    LIST
 5235
 5236
 5237 005420                            DOPINS  BXCT,164,T.NIL+T.SPOP
      000072      045                           .BYTE BITSX!T.OPFN
      000072      000                           .BYTE BITSY
      000244  000000G                           .WORD   D$BXCT
 5272 005420                                    LIST
 5273
 5274
 5275
 5276
 5277
 5278 005420                            MOPINS  BXIO,166,T.NIL+T.SPOP
      000073      045                           .BYTE BITSX!T.OPFN
      000073      000                           .BYTE BITSY
      000246  000000G                               .WORD   M$BXIO
 5287 005420                                    LIST
 5288
 5289
 5290 005420                            DOPINS  BXIO,166,T.NIL+T.SPOP
      000073      045                           .BYTE BITSX!T.OPFN
      000073      000                           .BYTE BITSY
      000246  000000G                           .WORD   D$BXIO
 5302 005420                                    LIST
 5303
 5304
 5305
 5306
 5307 005420                            MOPINS  BXLC,212,T.NIL
      000105      005                           .BYTE BITSX!T.OPFN
      000105      000                           .BYTE BITSY
      000272  000000G                               .WORD   M$BXLC
 5340 005420                                    LIST
 5341
 5342
 5343
 5344
 5345
 5346
 5347 005420                            MOPINS  BXPP,170,T.NIL+T.SPOP
      000074      045                           .BYTE BITSX!T.OPFN
      000074      000                           .BYTE BITSY
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 74-1
MIXED FUNCTION          QUOTE QUAD OUTPUT

        000250  000000G                         .WORD   M$BXPP
   5356 005420                                 LIST
   5357
   5358
   5359 005420                         DOPINS  BXPP,170,T.NIL+T.SPOP
        000074     045                         .BYTE BITSX!T.OPFN
        000074     000                         .BYTE BITSY
        000250  000000G                         .WORD   D$BXPP
   5381 005420                                 LIST
   5382
   5383
   5384
   5385
   5386
   5387
   5388 005420                         MOPINS  BXPW,172,T.NIL+T.SPOP
        000075     045                         .BYTE BITSX!T.OPFN
        000075     000                         .BYTE BITSY
        000252  000000G                         .WORD   M$BXPW
   5397 005420                                 LIST
   5398
   5399
   5400 005420                         DOPINS  BXPW,172,T.NIL+T.SPOP
        000075     045                         .BYTE BITSX!T.OPFN
        000075     000                         .BYTE BITSY
        000252  000000G                         .WORD   D$BXPW
 * 5417 005420                                 LIST
 * 5418
   5419
   5420
   5421
   5422
   5423 005420                         MOPINS  BXRL,174,T.NIL+T.SPOP
        000076     045                         .BYTE BITSX!T.OPFN
        000076     000                         .BYTE BITSY
        000254  000000G                         .WORD   M$BXRL
   5432 005420                                 LIST
   5433
   5434
   5435 005420                         DOPINS  BXRL,174,T.NIL+T.SPOP
        000076     045                         .BYTE BITSX!T.OPFN
        000076     000                         .BYTE BITSY
        000254  000000G                         .WORD   D$BXRL
   5447 005420                                 LIST
   5448
   5449
   5450
   5451
   5452
   5453 005420                         MOPINS  BXTS,200,T.NIL
        000100     005                         .BYTE BITSX!T.OPFN
        000100     000                         .BYTE BITSY
        000260  000000G                         .WORD   M$BXTS
   5562 005420                                 LIST
   5563
   5564
   5565
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 74-2
MIXED FUNCTION          QUOTE QUAD OUTPUT

   5566
   5567
   5568
   5569 005420                            MOPINS  BXWA,176,T.NIL
        000077      005                            .BYTE BITSX!T.OPFN
        000C77      000                            .BYTE BITSY
        000256  000000G                                .WORD   M$BXWA
   5592 005420                                    LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 75
MIXED FUNCTION          QUOTE QUAD OUTPUT

   5594 005420                          MOPINS  BXCR,206
        000103      001                         .BYTE BITSX!T.OPFN
        000103      000                         .BYTE BITSY
        000266  000000G                             .WORD   M$BXCR
   5716 005420                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 76
MIXED FUNCTION          QUOTE QUAD OUTPUT

   5718 005420                          MOPINS  BXEX,202
        000101      001                         .BYTE BITSX!T.OPFN
        000101      000                         .BYTE BITSY
        000262  000000G                             .WORD   M$BXEX
   5775 005420                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 77
MIXED FUNCTION          QUOTE QUAD OUTPUT

   5777 005420                          MOPINS  BXFX,210
        000104      001                         .BYTE BITSX!T.OPFN
        000104      000                         .BYTE BITSY
        000270  000000G                             .WORD   M$BXFX
   5879 005420                          LIST
```

```
 5881 005420                            MOPINS  BXNC,204
      000102     001                             .BYTE BITSX!T.OPFN
      000102     000                             .BYTE BITSY
      000264  000000G                                .WORD   M$BXNC
 5950 005420                                     LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 79
MIXED FUNCTION          QUOTE QUAD OUTPUT

   5952 005420                          MOPINS  BXNL,214
        000106     001                          .BYTE BITSX!T.OPFN
        000106     000                          .BYTE BITSY
        000274  000000G                             .WORD   M$BXNL
   5963 005420                                  LIST
   5964
   5965
   5966 005420                          DOPINS  BXNL,214
        000106     001                          .BYTE BITSX!T.OPFN
        000106     000                          .BYTE BITSY
        000274  000000G                         .WORD   D$BXNL
   6104 005420                                  LIST
```

```
 6106 005420                            DOPINS  FGET,150,T.DIM
      000064     101                             .BYTE BITSX!T.OPFN
      000064     000                             .BYTE BITSY
      000230  000000G                            .WORD   D$FGET
 6330 005420                                     LIST
 6331
 6332
 6333
 6334
 6335
 6336 005420                            DOPINS  FPUT,152,T.DIM
      000065     101                             .BYTE BITSX!T.OPFN
      000065     000                             .BYTE BITSY
      000232  000000G                            .WORD   D$FPUT
 6485 005420                                     LIST
```

```
  6487 005420                          DOPINS  FPNT,154,T.DIM
       000066     101                          .BYTE BITSX!T.OPFN
       000066     000                          .BYTE BITSY
       000234  000000G                         .WORD   D$FPNT
  6714 005420                                  LIST
```

```
 6716 005420                            MOPINS  1SCN,130
      000054     001                             .BYTE BITSX!T.OPFN
      000054     000                             .BYTE BITSY
      000210  000000G                                 .WORD   M$1SCN
 6723 005420                                     LIST
 6724
 6725
 6726 005420                            MOPINS  SCAN,62,T.DIM
      000031     101                             .BYTE BITSX!T.OPFN
      000031     000                             .BYTE BITSY
      000142  000000G                                 .WORD   M$SCAN
 6732 005420                                     LIST
 6733
 6734
 6735 005420                            MOPINS  1RDT,124
      000052     001                             .BYTE BITSX!T.OPFN
      000052     000                             .BYTE BITSY
      000204  000000G                                 .WORD   M$1RDT
 6741 005420                                     LIST
 6742
 6743
 6744 005420                            MOPINS  RDCT,60,T.DIM
      000030     101                             .BYTE BITSX!T.OPFN
      000030     000                             .BYTE BITSY
      000140  000000G                                 .WORD   M$RDCT
 7044 005420                                     LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 83
MIXED FUNCTION          QUOTE QUAD OUTPUT

   7046 005420                          MOPINS  GETIDN,0
   7193 000060                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 84
MIXED FUNCTION          QUOTE QUAD OUTPUT

   7195 000060                          MOPINS  SATAN,0
   7404 000060                                  LIST
```

```
 7406 000060                            MOPINS  OUTR,156
      000067     001                            .BYTE BITSX!T.OPFN
      000067     000                            .BYTE BITSY
      000236  000000G                               .WORD   M$OUTR
 7476 005420                                    LIST
```

```
 7478 005420                    MOPINS  DOUTER,0
 7504 000060                            LIST
 7505
 7506
 7507
 7508
 7509
 7510 000060                    MOPINS  PRDSET,0
 7564 000060                            LIST
 7565
 7566
 7567
 7568
 7569
 7570 000060                    MOPINS  INSPAC,0
 7612 000060                            LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 87
MIXED FUNCTION          QUOTE QUAD OUTPUT

   7614 000060                          DOPINS  INNR,156
        000067     001                          .BYTE BITSX!T.OPFN
        000067     C00                          .BYTE BITSY
        000236  000000G                         .WORD   D$INNR
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 88
MIXED FUNCTION          QUOTE QUAD OUTPUT

   7960 005420                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 89
MIXED FUNCTION          QUOTE QUAD OUTPUT

   7962 005420                          MOPINS  SUBR,0
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 90
MIXED FUNCTION          QUOTE QUAD OUTPUT

    8725 000060                                     LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 91
MIXED FUNCTION          QUOTE QUAD OUTPUT

   8727 000060                          MOPINS  RAND,0
   8771 000060                                  LIST
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 92
MIXED FUNCTION          QUOTE QUAD OUTPUT

      1         000001                              .END
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 92-1
SYMBOL TABLE

ACOPY = ****** GX        CHRDOT= 000056          CTEPS   000152R  025 D$BXIO= ****** G        D$$PLU  000016RG  020
AC0   =%000000           CHREP = 000105          CTLMLP  003512R  020 D$BXNL= ****** G        D$$RES  000530RG  020
AC1   =%000001           CHRERR= 104431          CTLMN   004300R  020 D$BXPP= ****** G        D$$SUB  000052RG  020
AC2   =%000002           CHRFIL= 000177          CTLMX   004520R  020 D$BXPW= ****** G        D$1COM= ****** G
AC3   =%000003           CHRFRO= 000135          CTLM0   003172R  020 D$BXRL= ****** G        D$1EXP= ****** G
AC4   =%000004           CHRLBR= 000073          CTLM1   004034R  020 D$CIRC= ****** G        D$1ROT= ****** G
AC5   =%000005           CHRLPR= 000072          CTMAX   000212R  025 D$COMB= ****** G        D.AND = 000000
ALLINT  003274R  025     CHRMNS= 000137          CTRMAX  000172R  025 D$COMP= ****** G        D.BXCT= 000007
ARYRET  003276R  025     CHRMUL= 000075          C.ALOC= 000162       D$CONT= ****** G        D.BXIO= 000007
ASCQUO= 000042           CHRNEG= 000100          C.BLOK= 000064       D$CTLM  003014RG  020  D.BXNL= 000007
A.ALEN= 000004           CHRPDL= 000271          C.BLSZ= 000140       D$DEAL= ****** G        D.BXPP= 000007
A.ARG = 000040           CHRPLU= 000055          C.BUFF= 177760       D$DECD= ****** G        D.BXPW= 000007
A.ASB = 000200           CHRPWR= 000120          C.BUFL= 177762       D$DIV   000154RG  020  D.BXRL= 000007
A.BITS= 000003           CHRQST= 000121          C.CCNT= 177766       D$DMD = ****** G        D.CIRC= 000001
A.CON = 000020           CHRQUO= 000113          C.CPTR= 177764       D$DROP= ****** G        D.COMB= 000001
A.DIMS= 000002           CHRRBR= 000047          C.CZ  = 001000       D$ENCD= ****** G        D.COMP= 000001
A.DIM1= 000006           CHRRPR= 000042          C.DIG = 000040       D$EQ    001060RG  020  D.CONT= 000006
A.NUM = 000006           CHRSEM= 000074          C.FASW= 020000       D$EXPD= ****** G        D.CTLM= 000000
A.PTR = 000000           CHRSMO= 000112          C.FLEN= 000010       D$FGET= ****** G        D.DEAL= 000003
A.TMP = 000010           CHRSPA= 000040          C.FLSH= 004000       D$FMT = ****** G        D.DECD= 000002
BADDIR= 104463           CHRUPA= 000131          C.FRC = 000001       D$FPNT= ****** G        D.DIV = 000000
BADERR= 104437           CHR.BS= 000010          C.IDEN= 000010       D$FPUT= ****** G        D.DMD = 000010
BITSX = 000000           CHR.CC= 000003          C.NAME= 000102       D$GE    001126RG  020  D.DQAD= 177777
BITSY = 000000           CHR.CR= 000015          C.NOBF= 000400       D$GT    001150RG  020  D.DROP= 000002
BLZBUB  005116R  025     CHR.CZ= 000032          C.NUM = 000020       D$IBM = ****** G        D.ENCD= 000002
BMONCT= ****** GX        CHR.GT= 000046          C.OP  = 000004       D$IND = ****** G        D.EQ  = 000000
BOPTAB  000000RG  005    CHR.LF= 000012          C.OPN = 010000       D$INNR= ****** G        D.EXPD= 000002
BRSELF  005046R  025     CHR.LT= 000043          C.RDLN= 177776       D$INTR= ****** G        D.FGET= 000005
CCNT    005072R  025     CHR.SI= 000017          C.REWT= 002000       D$LE    001036RG  020  D.FMT = 000004
CHANEX  003060R  025     CHR.SL= 000057          C.RND = 000010       D$LOG = ****** G        D.FPNT= 000005
CHAN00  000256R  025     CHR.SO= 000016          C.SBIT= 177756       D$LT    001014RG  020  D.FPUT= 000005
CHAN01  000330R  025     CHR.TB= 000011          C.SPFX= 000100       D$MAX   000324RG  020  D.GE  = 000000
CHAN02  000512R  025     CHR.UL= 000106          C.SPGL= 000200       D$MEM = ****** G        D.GT  = 000000
CHAN03  000674R  025     CHR..A= 000141          C.SPLX= 000001       D$MIN   000244RG  020  D.IBM = 000002
CHAN04  001056R  025     CHR..E= 000145          C.SPRS= 000002       D$MUL   000070RG  020  D.IND = 000002
CHAN05  001240R  025     CHR..S= 000163          C.SRC = 177774       D$NE    001172RG  020  D.INNR= 000013
CHAN06  001422R  025     CHR..T= 000164          C.SRCP= 000024       D$NN    000736RG  020  D.INTR= 000006
CHAN07  001604R  025     CHR..U= 000165          C.TT  = 000004       D$NR    000764RG  020  D.LE  = 000000
CHAN08  001766R  025     CHR..Z= 000172          C.WIDC= 177770       D$OR    000720RG  020  D.LOG = 000001
CHAN09  002150R  025     CHR..0= 000060          C.WIDR= 177772       D$PLUS  000000RG  020  D.LT  = 000000
CHAN10  002332R  025     CHR..1= 000061          DEBUG = 000000       D$PSUB= ****** G        D.MAX = 000000
CHAN11  002514R  025     CH.AND= 000001          DEFERR= 104405       D$PWR = ****** G        D.MEM = 000002
CHAN12  002676R  025     CMDBUF  003250R  025    DEVIL   005114R  025 D$QQAD  005364RG  020  D.MIN = 000000
CHAN13  000022R  023     CMDMAX= 000010          DEVNTF= 104470       D$QUAD  005150RG  020  D.MUL = 000000
CHAN14  000204R  023     CNTSIZ  005130R  025    DIGITS  000006R  025 D$RES   000404RG  020  D.NE  = 000000
CHNOPN= 104471           COMBIN  005076R  025    DIMAX = 000020       D$RHO   002350RG  020  D.NN  = 000000
CHRAND= 000051           COMCNT  005126R  025    DIMBLK  003112R  025 D$ROT = ****** G        D.NR  = 000000
CHRBEL= 000007           COMERR= 104425          DIMPTR  003114R  025 D$SCON= ****** G        D.OR  = 000000
CHRBKR= 000133           COMPTR  005122R  025    DIMTYP  003116R  025 D$SDIF= ****** G        D.PLUS= 000000
CHRBOL= 000200           COMRTN  003156R  025    DIRFUL= 104466       D$SUB   000034RG  020  D.PSUB= 000006
CHRBOX= 000114           COMSKP  005124R  025    DOMERR= 104417       D$SUBS= ****** G        D.PWR = 000001
CHRCMA= 000054           CORPTR  005120R  025    DPTERR= 104416       D$TAKE= ****** G        D.QQAD= 000000
CHRCOL= 000076           CS.ASS= 000200          DYDBTL  000000RG  007 D$TRAN= ****** G        D.QUAD= 000000
CHRDEL= 000107           CS.BOO= 000010          DYDCNT  003224R  025 D$UNON= ****** G        D.RES = 000000
CHRDIV= 000053           CS.FLO= 000020          DYDTAB  000060RG  004 D$$MAX  000350RG  020  D.RHO = 000000
CHRDLR= 000176           CS.IDN= 000040          D$AND   000672RG  020 D$$MIN  000270RG  020  D.ROT = 000003
CHRDLT= 000110           CS.ONE= 000100          D$BXCT= ****** G      D$$MUL  000126RG  020  D.SCON= 000006
```

SYMBOL TABLE

```
D.SDIF= 000006          FD.MNT= 100000          F.BITS= 177774          F.RTYP= 000000          LARGLE  003174R  025
D.SUB = 000000          FD.OSP= 004000          F.BKDN= 000026          F.R2  = 000014          LARGLN  003166R  025
D.SUBS= 000006          FD.PLC= 000004          F.BKDS= 000020          F.R3  = 000016          LARGPT  003164R  025
D.TAKE= 000002          FD.PRN= 000004          F.BKEF= 000050          F.R4  = 000020          LARGSF  003172R  025
D.TRAN= 000004          FD.PSE= 010000          F.BKP1= 000051          F.SEQN= 000100          LARGTY  003170R  025
D.UNON= 000006          FD.RAH= 000001          F.BKST= 000024          F.SPDV= 000072          LARGVA  003200R  025
D.$MAX= 000000          FD.RAN= 000002          F.BKVB= 000064          F.SPUN= 000074          LASTOP= 000120
D.$MIN= 000000          FD.REC= 000001          F.BR  = 000020          F.STBK= 000036          LBINC   005100R  025
D.$MUL= 000000          FD.RWM= 000001          F.CHR = 000075          F.STK = 000022          LBUFF   003316R  025
D.$PLU= 000000          FD.SDI= 000020          F.CNTG= 000034          F.STOP= 000004          LBUFLN= 000621
D.$RES= 000000          FD.SQD= 000040          F.CODE= 000010          F.STP = 000001          LCOUNT  005106R  025
D.$SUB= 000000          FD.TTY= 000004          F.DFNB= 000046          F.SUSP= 000002          LENERR= 104412
D.1COM= 000001          FD.WBH= 000002          F.DSPT= 000044          F.TO.I= ****** GX       LENTAB  000054R  025
D.1EXP= 000002          FF.CHR= 000005          F.DVNM= 000134          F.TRC = 000002          LEXPTR  003100R  025
D.1ROT= 000003          FF.NV = 000003          F.EFBK= 000010          F.UNIT= 000136          LFTOP   005066R  025
EDICHR  005063R  025    FF.POE= 000002          F.EFN = 000050          F.URBD= 000020          LINC    005102R  025
EDICNT  005064R  025    FF.RWD= 000001          F.END = 002000          F.VAL = 177776          LINE    005016R  025
EDIFLG  005062R  025    FF.RWF= 000006          F.EOBB= 000032          F.VBN = 000064          LININC  005026R  025
EDIT  = 000003          FF.SPC= 000004          F.ERR = 000052          F.VBSZ= 000060          LISIZE= 000020
EIS   = 000000          FHEADR  005012R  025    F.FACC= 000043          GETCOR= ****** GX       LITOP   005072R  025
ELESIZ  005132R  025    FHRET   005040R  025    F.FFBY= 000014          GETFRA= ****** GX       LNLNG = 104541
ENDCOR  000004R  025    FILNTF= 104467          F.FLAG= 000002          GETL  = ****** GX       LOCLCT  005036R  025
EOF   = 104475          FIS   = 000000          F.FLB = 000100          GETLIN= ****** GX       LOWCOR  000002R  025
ERRMAX= 000166          FISCON= 046000          F.FNAM= 000110          GETSM1= ****** GX       LSTERP  004766R  025
ERRNUM  004772R  025    FIXLIN= ****** GX       F.FNB = 000102          GETSM3= ****** GX       LWIDTH  000070R  025
ERRPTR  004770R  025    FIXPTR  003074R  025    F.FNDF= 000010          GNOME   003102R  025   L.BITS= 000003
ERRO    004774R  025    FLAGS   004764R  025    F.FTYP= 000116          HASHC   005136R  025   L.CNT = 000002
EXEBUF  005054R  025    FLCBUF  003250R  025    F.FUN = 000010          HASHTB  005140R  025   L.PRNT= 000001
EXEC  = ****** GX       FLINE   005014R  025    F.FVER= 000120          HSHMSK= 177601          L.PTR = 000000
E$CHR = ****** GX       FLOEXP  003270R  025    F.HEAD= 000006          HSHSIZ= 000200          L.PTR1= 000004
E$CT    005056R  025    FLONEG  003272R  025    F.HIBK= 000004          IASH  = ****** GX       L.SUBS= 000002
E$PT    005060R  025    FLSWTB  000222R  025    F.LBL = 000020          ICHAN   005050R  025   MACHIN= 000106
FA.APD= 000100          FLTLEN= 000010          F.LINE= 000012          IDIV  = ****** GX       MAXCHN= 000014
FA.CRE= 000010          FMONCT= ****** GX       F.LOC = 000004          ILFNAM= 104464          MONBLK  003120R  025
FA.DLK= 001000          FNINSI  005044R  025    F.LSTC= 000004          ILLCHN= 104540          MONBTL  000000RG 006
FA.ENB= 100000          FORIGN  000032R  025    F.LUN = 000042          ILLSW = 104565          MONCVT  003132R  025
FA.EXC= 002000          FO.APD= 000106          F.MBCT= 000054          IMONCT= ****** GX       MONIBM= ****** G
FA.EXT= 000004          FO.MFY= 000002          F.MBC1= 000055          IMUL  = ****** GX       MONLEN  003130R  025
FA.NSP= 000100          FO.RD = 000001          F.MBFG= 000056          INDERR= 104410          MONPTR  003122R  025
FA.NSR= 000200          FO.UPD= 000006          F.MON = 000002          INNTMP  003250R  025    MONSFT  003126R  025
FA.POS= 010000          FO.WRT= 000016          F.NIL = 000001          INNTM2  005016R  025    MONTAB  000060RG 003
FA.RD = 000001          FPMODE  000042R  025    F.NRBD= 000024          INPTR   000064R  025    MONTYP  003124R  025
FA.RWD= 004000          FPP   = 000000          F.NREC= 000030          INSNAM= 104462          MONVAL  003134R  025
FA.SEQ= 040000          FPPLEN= 000004          F.OPN = 000010          IOERR = 104477          MSABS   002022RG 020
FA.SHR= 000040          FREBLK  007400R  025    F.OVBS= 000030          I.CNT = 177776          MSBXAV= ****** G
FA.TMP= 000020          FRESCT= ****** GX       F.PREV= 000000          I.DIM = 177772          MSBXCR= ****** G
FA.WCK= 020000          FUZZ    000142R  025    F.PRNT= 100000          I.LEN = 000012          MSBXCT= ****** G
FA.WRT= 000002          FXPTR1  003076R  025    F.PROT= 010000          I.OERR= 000062          MSBXEX= ****** G
FD.BLK= 000010          F$EVAL= 000100          F.RACC= 000016          I.OLD = 177770          MSBXFX= ****** G
FD.CCL= 000002          F$EVI = 000200          F.RATT= 000001          I.PTR = 177774          MSBXIO= ****** G
FD.COM= 020000          F$EXEC= 000400          F.RCNM= 000034          I.STR = 177766          MSBXLC= ****** G
FD.CR = 000002          F$FUNC= 001000          F.RCTL= 000017          LABERR= 104406          MSBXNC= ****** G
FD.DIR= 000010          F.ACTL= 000076          F.READ= 004000          LABLPT  005042R  025   MSBXNL= ****** G
FD.FTN= 000001          F.ALOC= 000040          F.REDF= 000040          LARGBL  003162R  025   MSBXPP= ****** G
FD.F11= 040000          F.BBFS= 000062          F.RES = 000040          LARGCT= ****** GX       MSBXPW= ****** G
FD.INS= 000010          F.BDB = 000070          F.RET = 177772          LARGCV  003176R  025   MSBXRL= ****** G
FD.ISP= 002000          F.BGBC= 000057          F.RSIZ= 000002          LARGFL  003160R  025   MSBXTS= ****** G
```

APLOP MACRO M1113 08-FEB-80 09:04 PAGE 92-3
SYMBOL TABLE

```
MSBXWA= ****** G        M.DOUT= 000006        NB.SNM= 000040        OPTMP1  005070R  025 RTOP    005070R  025
MSDQAD= ****** G        M.DQAD= 000006        NB.STP= 000020        OPTMP2  005072R  025 RT11  = 000000
MSENCD= ****** G        M.ENCD= 000004        NB.SVR= 000010        OPTMP3  005074R  025 R.BITS= 000003
MSEXE   004644RG  020   M.EXE = 000000        NB.TYP= 000002        OPTMP4  005076R  025 R.FIX = 000001
MSEXP = ****** G        M.EXP = 000001        NB.VER= 000001        OPTMP5  005100R  025 R.LCT = 000002
MSFACT= ****** G        M.FACT= 000001        ND.DEV= 000200        OPTMP6  005102R  025 R.LN  = 000002
MSFMT = ****** G        M.FIL = 000177        ND.EXT= 000002        OPTMP7  005104R  025 R.LNUM= 000010
MSGDUP= ****** G        M.FLR = 177777        ND.NAM= 000004        OPTMP8  005106R  025 R.NEXT= 000006
MSGDWN= ****** G        M.FMT = 000004        ND.PPN= 000100        OPTMP9  005110R  025 R.OFF = 000000
MSIDN   001240RG  020   M.GDUP= 000003        ND.$EX= 000000        OPTM10  005112R  025 R.PREV= 000004
MSIOTA  004522RG  020   M.GDWN= 000003        ND.$E$= 000020        OPTM14  005026R  025 R.PTR = 000000
MSLN  = ****** G        M.GETI= 000006        ND.$NM= 000000        ORIGIN  000030R  025 R.SEQ = 000003
MSMMD = ****** G        M.IBM = 000003        ND.$N$= 000040        OUTPTR  000066R  025 R.VAR = 000002
MSNEG   001616RG  020   M.IDN = 000000        ND.$PN= 001400        OVROPR= 000000        R6    =%000006
MSNOT   002056RG  020   M.INSP= 000006        NEGSGN  005063R  025 PATCH   007344R  025 R7    =%000007
MSNUB = ****** G        M.IOTA= 000000        NEWLIN= ****** GX     PI      000000RG  002 SAME  = ****** GX
MSOUTR= ****** G        M.LIT = 000205        NINPT1  003302R  025 PRECSN  000052R  025 SCALOP= 000030
MSPITM  002002RG  020   M.LN  = 000001        NINPT2  003300R  025 PRINT = ****** GX     SI    =%000005
MSQQAD  005224RG  020   M.MMD = 000010        NINSTK  003304R  025 RANDAC  003064R  025 SIDAMN= 104433
MSQUAD  005026RG  020   M.NEG = 000000        NINTMP  003310R  025 RANDY   003062R  025 SIFRAM  003314R  025
MSRAVL  002762RG  020   M.NOT = 000000        NINTYP  003306R  025 RANERR= 104411        SISTAK  006350R  025
MSRCIP  001744RG  020   M.NUB = 000006        NOTAVL= 104472        RARGBL  003204R  025 SITOP   006346R  025
MSRDCT= ****** G        M.NUM = 000204        NOTOPN= 104473        RARGCT= ****** GX     SITP    007144R  025
MSREVS= ****** G        M.OUTR= 000012        NPRT1   004762R  025 RARGCV  003220R  025 SIWARN= 000200
MSROLL  002166RG  020   M.PITM= 000000        NULL    007364R  025 RARGFL  003202R  025 SIZTAB  000044R  025
MSSCAN= ****** G        M.PRDS= 000006        NUMCHN= 000020        RARGLE  003216R  025 SI.BIT= 177774
MSSHAP  002266RG  020   M.QQAD= 000000        NUMDFL  003230R  025 RARGLN  003210R  025 SI.LEN= 177772
MSSIGN  001650RG  020   M.QUAD= 000000        NUMEFL  003233R  025 RARGPT  003206R  025 SI.RET= 177772
MSTRAN= ****** G        M.RAND= 000012        NUMESN  003236R  025 RARGSF  003214R  025 SI.VAL= 177776
MSSABS  002040RG  020   M.RAVL= 000000        NUMFLG  003230R  025 RARGTY  003212R  025 SKPSIZ  005134R  025
MSSCEI  001560RG  020   M.RCIP= 000000        NUMLEN  000062R  025 RARGVA  003222R  025 SM.BB = ****** GX
MSSFLR  001260RG  020   M.RDCT= 000011        NUMNEG  003232R  025 RBASE   005074R  025 SPHOLD  000004R  024
MSSIDN  001250RG  020   M.REVS= 000003        NUMQFL  003234R  025 RCOUNR  005112R  025 SPINIT  000002R  024
MSSNEG  001632RG  020   M.ROLL= 000000        NUMRIG  003231R  025 RCOUNT  005110R  025 STKINC= 000774
MSSSIG  001676RG  020   M.SATA= 000001        NUMWID  003226R  025 REDCNT  005070R  025 STKXTN= ****** GX
MS1RDT= ****** G        M.SCAN= 000011        NUMZFL  003235R  025 REDIGN  005066R  025 STOP  = 104420
MS1REV= ****** G        M.SHAP= 000000        NYIERR= 104432        REDOFF  005072R  025 SUBCNT  003104R  025
MS1SCN= ****** G        M.SIGN= 000000        N.BLSZ= 000036        REENTR  003060R  025 SUBDCT  003106R  025
M.ABS = 000000          M.SUBR= 000001        N.DESC= 000020        REGTYP= 000000        SUBPTR  003110R  025
M.BOL = 000200          M.SYM = 000203        N.DEV = 000032        RESBIT  003142R  025 SYMB1   005344R  025
M.BXAV= 000007          M.SYN = 000201        N.DEVN= 000034        RESCT = ****** GX     SYMF    005340R  025
M.BXCR= 000007          M.TRAN= 000012        N.DID = 000024        RESCVT  003152R  025 SYMPT1  005346R  025
M.BXCT= 000007          M.$ABS= 000000        N.DVNM= 000032        RESERR= 104440        SYMTAB  005352R  025
M.BXEX= 000007          M.$CEI= 000000        N.EXT = 000014        RESLEN  003150R  025 SYMTBF= 104404
M.BXFX= 000007          M.$FLR= 000000        N.FID = 000000        RESPTR  003140R  025 SYMTOP  005350R  025
M.BXIO= 000007          M.$IDN= 000000        N.FNAM= 000006        RESSFT  003146R  025 SYMTP   006272R  025
M.BXLC= 000003          M.$NEG= 000000        N.FTYP= 000014        RESTYP  003144R  025 SYMWRN= 000054
M.BXNC= 000007          M.$SIG= 000000        N.FVER= 000016        RESULT  003136R  025 SYNER  = ****** GX
M.BXNL= 000007          M.1RDT= 000011        N.NAME= 000006        RESVAL  003154R  025 SYNERR= 104407
M.BXPP= 000007          M.1REV= 000003        N.NEXT= 000022        RINC    005104R  025 SYSER  = ****** GX
M.BXPW= 000007          M.1SCN= 000011        N.PPN = 000024        RLTBAS  005066R  025 SYSERR= 104402
M.BXRL= 000007          NAMBUF  003240R  025  N.STAT= 000020        ROROX   002760R  020 SYSTAT  000000R  024
M.BXTS= 000007          NB.DEV= 000200        N.UNIT= 000034        ROTINC  005070R  025 SY.BAD= 000001
M.BXWA= 000007          NB.DIR= 000100        OBUFLN= 001000        ROUND   000010R  025 SY.BRK= 100000
M.CFL = 177777          NB.NAM= 000004        OCHAN   005052R  025 ROVER   003312R  025 SY.BS  = 000010
M.CHR = 000202          NB.SD1= 000400        OMODE   000402R  023 RSTS  = 000000        SY.CLR= 000040
M.CMT = 000206          NB.SD2= 001000        OPTMP0  005066R  025 RSX   = 000115        SY.ERP= 000020
```

```
APLOP   MACRO M1113  08-FEB-80 09:04  PAGE 92-4
SYMBOL TABLE

SY.NOU= 000100          S.LEN  = 000002          T.LIST= 000400          WIDTH   000072R     025 $NEGF  = ****** GX
SY.OPN= 000002          S.LINK= 177774           T.LIT  = 000020          WROUND  000020R     025 $RSTS  = ****** G
SY.PMT= 001000          S.NFEN= 000020           T.MON  = 000010          WSDATE  000136R     025 $RSX   = ****** G
SY.SRT= 000400          S.TYPE= 140000           T.NIL  = 000004          WSFULL= 104403          $RT11  = ****** G
SY.SVC= 000004          S.VAL  = 000000          T.NUM  = 002000          WSNAME  000074R     025 $STCFI= ****** GX
SY.SVL= 000200          TKDIMS  005072R     025  T.OPFN= 000001          WSTIME  000132R     025 $STF   = ****** GX
S.AB   = 177776         TKFRAM  005066R     025  T.OTHR= 040000          $ABSF  = ****** GX      $SUBF  = ****** GX
S.ASB  = 100000         TKLEFT  005070R     025  T.SOP  = 000002          $ADDF  = ****** GX      $TSTF  = ****** GX
S.BFHD= 000020          TMODE   000403R     023  T.SPOP= 000040          $CFCC  = ****** GX      $$$$$$= 005420R     020
S.BITS= 000002          TMPTAG= 000166           T.TMP  = 000200          $CLRF  = ****** GX      .BOOLE= 000002
S.CHR1= 000003          TRUFLR  001522R     020  T.VAL  = 100000          $CMPF  = ****** GX      .CHARA= 000000
S.FATT= 000016          TTYBUF  004140R     025  T.VAR  = 001000          $DIVF  = ****** GX      .DEF.  = 000002
S.FDB  = 000140         TTYLEN= 000622           UPPER  = 177400          $FIS   = ****** G       .FLOAT= 000006
S.FLB  = 040000         TURKEY= 104436           UTNAME  000344R     023  $FPP   = ****** G       .FSRCB= ****** G
S.FNAM= 000006          TYPE   = 000166          VALERR= 104413           $FPPLN= ****** G        .INTEG= 000004
S.FNB  = 000036         T.BOL  = 020000          VARBLE  007400R     025  $LDF   = ****** GX      .MOLUN  000404R     023
S.FNBW= 000017          T.CON  = 004000          VARBLS  000000R     025  $MACHN= ****** G       .RAND  = ****** GX
S.FNTY= 000004          T.DIM  = 000100          VERNUM  000000R     025  $MODF  = ****** GX      ...GBL= 000000
S.FTYP= 000002          T.FUN  = 010000          VRSION= 000002           $MULF  = ****** GX      ...TPC= 000020

. ABS.  000000      000
        000000      001
TABLES  000010      002
MONTB   000320      003
DYDTB   000320      004
BOPTB   000120      005
MONBTB  000107      006
DYDBTB  000120      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    005420      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  14969 WORDS  ( 59 PAGES)
DYNAMIC MEMORY:  16590 WORDS  ( 63 PAGES)
ELAPSED TIME:  00:05:16
APLOP,DM1:[305,301]APLOP/-SP=APLRX1,APLDEF,APLOP,END
```

FILEID**APLOO1

```
  AAAAAA      PPPPPPPP      LL              000000        000000          11
  AAAAAA      PPPPPPPP      LL              000000        000000          11
AA      AA    PP      PP    LL            00    00      00    00      1111
AA      AA    PP      PP    LL            00    00      00    00      1111
AA      AA    PP      PP    LL            00    00      00  0000        11
AA      AA    PP      PP    LL            00    00      00  0000        11
AA      AA    PPPPPPPP      LL            00    00      00  00  00      11
AA      AA    PPPPPPPP      LL            00    00      00  00  00      11
AAAAAAAAAA    PP            LL            00    00      0000    00      11
AAAAAAAAAA    PP            LL            00    00      0000    00      11
AA      AA    PP            LL            00    00      00      00      11          ....
AA      AA    PP            LL            00    00      00      00      11          ....
AA      AA    PP            LLLLLLLLLL      000000        000000        111111      ....
AA      AA    PP            LLLLLLLLLL      000000        000000        111111      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 09:10
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    47-   327    SCALAR FUNCTION         POWER
```

APL001 MACRO M1113 08-FEB-80 09:10
TABLE OF CONTENTS

47- 396	SCALAR FUNCTION	LOGARITHM
47- 421	SCALAR FUNCTION	CIRCULAR FUNCTIONS
47- 620	SCALAR FUNCTION	COMBINATION
49- 931	SCALAR FUNCTION	EXPONENTIAL
49- 943	SCALAR FUNCTION	NATURAL LOGARITHM
49- 996	SCALAR FUNCTION	FACTORIAL
54- 1702	MIXED FUNCTION	FIRST CO-ORDINATE COMPRESS
54- 1716	MIXED FUNCTION	COMPRESSION
86- 7196	SATAN	GENERAL PURPOSE DATA MOVER
91- 7963	SCIENTIFIC SUBROUTINES	
91- 7965	SUBROUTINE	CLOSE0: Fuzzily close to zero?
91- 7983	SUBROUTINE	.COS: COSINE
91- 7999	SUBROUTINE	.SIN: SINE
91- 8093	SUBROUTINE	.TAN: TANGENT
91- 8111	SUBROUTINE	.LN: NATURAL LOGARITHM
91- 8197	SUBROUTINE	.EXP: EXPONENTIAL
91- 8307	SUBROUTINE	.ACOS
91- 8324	SUBROUTINE	.ASIN: ARCSINE
91- 8359	SUBROUTINE	.ATAN: ARCTANGENT
92- 8465	SUBROUTINE	.SINH: HYPERBOLIC SINE
92- 8486	SUBROUTINE	.COSH: HYPERBOLIC COSINE
92- 8507	SUBROUTINE	.TANH: HYPERBOLIC TANGENT
92- 8540	SUBROUTINE	.ATANH: HYPERBOLIC ARCTANGENT
92- 8561	SUBROUTINE	.ACOSH: HYPERBOLIC ARCCOSINE
92- 8580	SUBROUTINE	.ASINH: HYPERBOLIC ARCSINE
92- 8598	SUBROUTINE	.FACT: GENERALIZED FACTORIAL

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 1


      1        000000                RT11    =  0          ;NOT RT-11
      2        000000                RSTS    =  0          ;NOT RSTS/E
      3        000115                RSX     = 'M          ;RSX-11/M
      4
      5        000000                EIS     =  0          ;NO EIS
      6        000000                FIS     =  0          ;NO FIS
      7        000000                FPP     =  0          ;NO FPP
      8
      9        000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOO1  MACRO M1113  08-FEB-80 09:10  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  APLOO1
                                        .IDENT  \V02.03\
                                        ;
                                        ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
                                        ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
                                        ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
                                        ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
                                        ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO1 IS:
                                        ;
                                        ;       APLOO1 < ...,APLOO1,APLOP
                                        ;
                                        ;
000001                                  OVROPR= 1                       ;OVERLAY SPECIFIED OPERATORS
```

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                                ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                                ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                                ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                                ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                                ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                                ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                                ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                                ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                                ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                                ;       INTACT!
                                ;--



                                ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                                ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                                ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                                ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                                ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                                ;MACROS.

                                .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-D.'NAME
                                        .PSECT  CODE
                                D$'NAME::
                                .ENDM

                                .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-M.'NAME
                                        .PSECT  CODE
                                        .IF NE NUM
                                M$'NAME::
                                    .ENDC
                                .ENDM

                                .MACRO  LIST
                                        .ENDC
                                .ENDM

                                .MACRO  PRMRES
                                        .IF EQ 1
                                .ENDM

                                        ;********** ASSEMBLING THE FILE!!!!!
007400                          PRMRES
                                        LIST
```

```
                                        DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
                                                LIST

                                        DOPINS  $PLU,-2
                                                LIST

                                        DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
                                                LIST

                                        DOPINS  $SUB,-4
                                                LIST


                                        DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
                                                LIST


                                        DOPINS  $MUL,-6
                                                LIST

                                        DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
                                                LIST


                                        DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
                                        .SBTTL  SCALAR FUNCTION          POWER

000000                                          STF     AC2,AC0         ;COPY EXPONENT
000006                                          CALL    CLOSE0          ;IS IT (+-)X^0 ?
000012  001005                                  BNE     1$              ;NO
000014                                          LDF     #^F1,AC0        ;YES, RESULT=1
000024                                          RETURN

000026                                  1$:     STF     AC1,AC0         ;COPY BASE
000034                                          CALL    CLOSE0          ;IS IT 0^(+-)X?
000040  001013                                  BNE     3$              ;NO
000042                                          TSTF    AC2             ;IS IT 0^-X?
000050                                          CFCC                    ;WELL, I'M WAITING
000054  100001                                  BPL     2$              ;NO
000056  104417                          8$:     DOMERR                  ;DOMAIN ERROR
000060                                  2$:     CLRF    AC0             ;0^X=0
000066                                          RETURN

000070                                  3$:     PUSH    R0              ;Save R0
000072                                          STF     AC1,AC3         ;SAVE BASE
000100                                          STF     AC2,AC0         ;COPY EXPONENT
000106                                          MODF    #^F1.,AC0       ;INTEGERIZE
000116                                          CALL    CLOSE0          ;ROUGHLY AN INTEGER ?
000122  001052                                  BNE     4$              ;NO
000124                                          STCFI   AC1,R0          ;CONVERT TO INTEGER
000132  103446                                  BCS     4$              ;OVERFLOW, GIVE UP
000134  002001                                  BGE     9$              ;POSITIVE #?
000136  005400                                  NEG     R0              ;NO, SO MAKE IT SO
000140                                  9$:     LDF     AC3,AC1         ;RESTORE AC1
000146                                          LDF     #^F1.,AC0       ;INITIAL VALUE OF 1
000156  006200                          10$:    ASR     R0              ;SHIFT TO GET LOW BIT
000160  103003                                  BCC     11$             ;NOTHING LOW
```

```
APLO01  MACRO M1113  08-FEB-80 09:10  PAGE 47-1
SCALAR FUNCTION          POWER

 359 000162                          MULF    AC3,AC0         ;MULTIPLY BY ...
 360 000170  005700          11$:    TST     R0              ;'CAUSE EMULATOR HAS PROBLEM
 361 000172  001404                  BEQ     12$             ;NOTHING LEFT
 362 000174                          MULF    AC3,AC3         ;SQUARE EXP
 363 000202  000765                  BR      10$             ;AND LOOP
 364 000204                  12$:    TSTF    AC2             ;CHECK SIGN OF EXPONENT
 365 000212                          CFCC
 366 000216  002012                  BGE     13$             ;NON-NEGATIVE
 367 000220                          LDF     #^F1.,AC3       ;LOAD A 1. INTO AC3
 368 000230                          DIVF    AC0,AC3         ;INVERT THE RESULT
 369 000236                          LDF     AC3,AC0         ;AND PUT IT BACK
 370 000244                  13$:    POP     R0              ;Restore R0
 371 000246                          RETURN
 372
 373 000250                  4$:     STF     AC3,AC1         ;GET AC1 BACK
 374 000256                          STF     AC2,AC0         ;COPY EXPONENT
 375 000264                          SUBF    #^F.5,AC0       ;SQRT(X)?
 376 000274                          CALL    CLOSE0          ;FUZZ IT AGAIN
 377 000300  001006                  BNE     5$              ;NO
 378 000302                          STF     AC1,AC0         ;COPY BASE
 379 000310                          CALL    .SQRT           ;GET SQUARE ROOT
 380 000314  000753                  BR      13$             ;And return
 381
 382 000316                  5$:     TSTF    AC1             ;TEST BASE
 383 000324                          CFCC                    ;X<0?
 384 000330  100652                  BMI     8$              ;YES
 385 000332                          STF     AC1,AC0         ;X^Y=EXP(Y*LN(X))
 386 000340                  6$:     CALL    .LN
 387 000344                          MULF    AC2,AC0
 388 000352                          CALL    .EXP
 389 000356  103332                  BCC     13$             ;Return if no overflow
 390 000360  104417                  DOMERR                  ;Overflow, Domain error
 392 000362                          LIST
 393
 394
 395 000362                  DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
 396                         .SBTTL  SCALAR FUNCTION         LOGARITHM
 397
 398 000362                          STF     AC1,AC0         ;COPY BASE
 399 000370                          CALL    .LN             ;GET IT'S NATURAL LOG
 400 000374                          STF     AC0,AC4         ;SAVE IT FOR LATER
 401 000402                          CFCC                    ;CHECK THE SIGN OF THE BASE'S LOG
 402 000406  001015                  BNE     1$              ;GOOD IF BASE ISN'T 1
 403
 404 000410                          CMPF    #^F1,AC2        ;SEE IF LOG BASE 1 OF 1
 405 000420                          CFCC
 406 000424  001401                  BEQ     2$              ;IF SO, POSSIBLE TO HAVE BASE=1
 407 000426  104417                  DOMERR                  ;OTHERWISE WE CAN'T DO IT
 408
 409 000430                  2$:     LDF     #^F1,AC0        ;LOG BASE 1 OF 1 IS 1 SINCE 1^1=1
 410 000440                          RETURN
 411
 412 000442                  1$:     STF     AC2,AC0         ;COPY THE NUMBER TO TAKE LOG OF
 413 000450                          CALL    .LN             ;FIND IT'S LOG
 414 000454                          DIVF    AC4,AC0         ;DO A BASE CONVERSION
 415 000462                          RETURN                  ;    LOG B (A)=LN(A)/LN(B)
 417 000464                          LIST
```

```
                                        DOPINS  CIRC,52,T.SOP,,FLOAT
                                        .SBTTL  SCALAR FUNCTION          CIRCULAR FUNCTIONS

                                                PUSH    R0               ;Save R0!
                                                STF     AC1,AC0          ;COPY AC1 => AC0
                                                CALL    F.TO.I           ;CONVERT TO INTEGER
000500  104417                          1$:     DOMERR                   ;WON'T CONVERT, ERROR
000502  013700  003154'                         MOV     RESVAL,R0        ;GET THE RESULT
000506                                          STF     AC2,AC0          ;SAVE NUMBER TO DO IT TO IN SCRATCH AC
000514  162700  000010                         SUB     #10,R0           ;SEE IF THE TYPE CIRC. FUN. IS LEGAL
000520  002367                              BGE     1$               ;ERROR, DIE
000522  062700  000017                      ADD     #17,R0           ; (THIS ACTUALLY WORKS, TRYIT SOMETIME)
000526  002764                              BLT     1$               ;ERROR, DIE
000530  006300                              ASL     R0               ;GET READY FOR WORD BRANCH IN TABLE
000532                                      CALL    @CIRTAB(R0)      ;Go to the user supplied function
000536                                      POP     R0               ;Restore R0
000540                                      RETURN

000000                                          .PSECT  TABLES

000000  004140'                      CIRTAB: .WORD   .ATANH           ;HYPERBOLIC ARCTANGENT
000002  004226'                              .WORD   .ACOSH           ;HYPERBOLIC ARCCOSINE
000004  004272'                              .WORD   .ASINH           ;HYBERBOLIC ARCSINE
000006  000542'                              .WORD   CNEG4
000010  000622'                              .WORD   .ATAN            ;ARCTANGENT
000012  000366'                              .WORD   .ACOS            ;ARCCOSINE
000014  000430'                              .WORD   .ASIN            ;ARCSINE
000016  000566'                              .WORD   CZERO
000020  002270'                              .WORD   .SIN             ;SINE
000022  002254'                              .WORD   .COS             ;COSINE
000024  002734'                              .WORD   .TAN             ;TANGENT
000026  000574'                              .WORD   CPOS4
000030  001526'                              .WORD   .SINH            ;HYPERBOLIC SINE
000032  001610'                              .WORD   .COSH            ;HYPERBOLIC COSINE
000034  001672'                              .WORD   .TANH            ;HYPERBOLIC TANGENT

000542                                          .PSECT  CODE

                                                .ENABL LSB


                                ;DO 'SQR(B^2-1)' ON CIRCULAR FUNCTION #-4
000542                          CNEG4:  MULF    AC2,AC0          ;SQUARE AC0
000550                                  SUBF    #^F1,AC0         ;GET (B^2)-(1)
000560                          2$:     CALL    .SQRT            ;FIND THE SQUARE ROOT OF THE WHOLE MESS
000564                                  RETURN                   ;RETURN WITH THE RESULT

                                ;DO 'SQR(1-B^2)' ON CIRCULAR FUNCTION #0
000566                          CZERO:  NEGF    AC0              ;NEGITIVE TIMES POSITIVE IS NEGITIVE

                                ;DO 'SQR(1+B^2)' ON CIRCULAR FUNCTION #4
000574                          CPOS4:  MULF    AC2,AC0          ;GET (+-)B^2
000602                                  ADDF    #^F1,AC0         ;ADD ONE TO THAT
000612  000762                          BR      2$               ;AND GO TO COMMON RETURN
```

```
                                        .DSABL LSB
000614                                  LIST


                        DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
000614                          LIST

                        DOPINS  $MIN,-14
000614                          LIST


                        DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
000614                          LIST

                        DOPINS  $MAX,-16
000614                          LIST


                        DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
000614                          LIST


                        DOPINS  $RES,-12
000614                          LIST


000614                  DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
                        .SBTTL  SCALAR FUNCTION         COMBINATION

000614                          STF     AC2,AC0         ;COMBINATION 'A!B' = A!/((B!)(A-B)!)
000622                          SUBF    AC1,AC0         ;GET A-B
000630                          CFCC                    ;CHECK ARGUMENTS
000634  100004                  BPL     1$              ;A <= B IS LEGAL
000636                          CLRF    AC0             ;ELSE NOT POSSIBLE TO COMBINE!
000644  000425                  BR 2$

000646                  1$:     CALL    .FACT           ;FIND IT'S FACTORIAL
000652                          STF     AC0,AC3         ;SAVE RESULTS FOR LATER
000660                          STF     AC1,AC0         ;COPY B
000666                          CALL    .FACT           ;AND FIND IT'S FACTORIAL
000672                          MULF    AC0,AC3         ;MULTIPLY TO GET THE DIVISOR
000700                          STF     AC2,AC0         ;COPY A
000706                          CALL    .FACT           ;AND GET IT'S FACTORIAL
000712                          DIVF    AC3,AC0         ;DIVIDE TO GET THE FINAL RESULT
000720                  2$:     RETURN
000722                          LIST
```

```
    641 000722                          DOPINS  AND,42,T.SOP,,BOOLEAN+CS.IDN+CS.ONE+CS.ASS
    652 000722                                  LIST
    653
    654
    655 000722                          DOPINS  OR,40,T.SOP,,BOOLEAN+CS.IDN+CS.ASS
    662 000722                                  LIST
    663
    664
    665 000722                          DOPINS  NN,46,T.SOP,,BOOLEAN
    676 000722                                  LIST
    677
    678 000722                          DOPINS  NR,44,T.SOP,,BOOLEAN
    687 000722                                  LIST
    688
    689
    690 000722                          DOPINS  LT,24,T.LIT+T.SOP,,CHARACTER+CS.IDN
    700 000722                                  LIST
    701
    702 000722                          DOPINS  LE,26,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
    712 000722                                  LIST
    713
    714
    715 000722                          DOPINS  EQ,30,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
    733 000722                                  LIST
    734
    735 000722                          DOPINS  GE,36,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
    745 000722                                  LIST
    746
    747
    748 000722                          DOPINS  GT,32,T.LIT+T.SOP,,CHARACTER+CS.IDN
    758 000722                                  LIST
    759
    760
    761 000722                          DOPINS  NE,34,T.LIT+T.SOP,,CHARACTER+CS.IDN
    780 000722                                  LIST
    781
```

B 1
C 1 ACOPY
D 1
E 1
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1 APLDX1
B 2
C 2
D 2
E 2
F 2
G 2
H 2
I 2
J 2
K 2
L 2
M 2
N 2
B 3
C 3
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5
I 5

J 5
K 5
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6 APLEM
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7
N 7
B 8
C 8
D 8
E 8
F 8
G 8
H 8
I 8
J 8
K 8
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9 APLOP
J 9
K 9
L 9
M 9
N 9
B 10
C 10
D 10

E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14

M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15 APL001
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16

```
000722                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,MSIDN
000722                                  LIST

000722                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,MSIDN
000722                                  LIST

000722                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
000722                                  LIST


000722                          MOPINS  $IDN,-2
000722                                  LIST


000722                          MOPINS  $FLR,-14
000722                                  LIST


000722                          MOPINS  $CEI,-16
000722                                  LIST


000722                          MOPINS  NEG,4,T.SOP,,INTEGER
000722                                  LIST

000722                          MOPINS  $NEG,-4
000722                                  LIST


000722                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
000722                                  LIST

000722                          MOPINS  $SIG,-6
000722                                  LIST


000722                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
000722                                  LIST

000722                          MOPINS  EXP,20,T.SOP,,FLOAT
                                .SBTTL  SCALAR FUNCTION                 EXPONENTIAL

000722                                  STF     AC1,AC0                 ;COPY ARG
000730                                  CALL    .EXP                    ;FIND THE EXP OF IT
000734  103001                          BCC     1$                      ;Check for overflow
000736  104417                          NOMERR                          ;There was overflow, Domain error
000740                          1$:     RETURN                          ;EASY!
000742                                  LIST


000742                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
                                .SBTTL  SCALAR FUNCTION                 NATURAL LOGARITHM

000742                                  STF     AC1,AC0                 ;COPY ARG
000750                                  CALL    .LN                     ;FIND LOG OF IT
000754                                  RETURN
000756                                  LIST
```

```
 950
 951
 952 000756                     MOPINS  PITM,52,T.SOP,,FLOAT
 959 000756                             LIST
 960
 961 000756                     PRMRES
 966 000756                             LIST
 967
 968
 969 000756                     MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
 977 000756                             LIST
 978
 979 000756                     MOPINS  $ABS,-12
 992 000756                             LIST
 993
 994
 995 000756                     MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
 996                            .SBTTL  SCALAR FUNCTION          FACTORIAL
 997
 998 000756                             STF     AC1,AC0          ;COPY ARG
 999 000764                             CALL    .FACT            ;GET IT'S FACTORIAL
1000 000770                             RETURN
1002 000772                             LIST
1003
1004
1005 000772                     MOPINS  NOT,50
1031 000772                             LIST
1032
1033
1034 000772                     MOPINS  ROLL,100
1057 000772                             LIST
```

```
   1059 000772                          DOPINS  DEAL,100
   1187 000772                                  LIST
```

SCALAR FUNCTION FACTORIAL

```
1189 000772                          MOPINS  SHAP,102
1211 000772                                  LIST
1212
1213
1214
1215
1216
1217 000772                          DOPINS  RHO,102
1313 000772                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 52
SCALAR FUNCTION          FACTORIAL

   1315 000772                          MOPINS  RAVL,56
   1325 000772                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 000772                          DOPINS  CTLM,56,T.DIM
   1596 000772                                  LIST
```

```
   1598 000772                          MOPINS  IOTA,72
   1632 000772                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 000772                          DOPINS  IND,72
   1699 000772                                  LIST
```

```
                                DOPINS  1COM,124
                                .SBTTL  MIXED FUNCTION          FIRST CO-ORDINATE COMPRESS

000772                                  CALL    GETREG          ;LOAD SOME REGISTERS
000776  012702  000001                  MOV     #1,R2           ;USE DIM 1
001002                                  CALL    FIXDIM          ;FIX THE DIM
001006  000137  001016'                 JMP     COMP1           ;DO THE COMPRESS
001012                                  LIST


                                DOPINS  COMP,60,T.DIM
                                .SBTTL  MIXED FUNCTION          COMPRESSION

001012                                  CALL    DFXDIM          ;GET A POINTER TO THE DIM
001016  013701  003170'         COMP1:  MOV     LARGTY,R1       ;GET LEFT ARG TYPE
001022  001001                          BNE     1$              ;YES
001024  104417                          DOMERR                  ;YOU TURKEY, YOU CAN'T COM. WITH LIT'S
001026  016137  000000G 003176' 1$:     MOV     BLARGC(R1),LARGCV ;CONVERT LEFT ARG TO BOOLEAN
001034  126327  000002  000001          CMPB    A.DIMS(R3),#1   ;IS LEFT ARG A VECTOR THEN
001042  001415                          BEQ     5$              ;YES, THAT'S GOOD
001044  002401                          BLT     2$              ;;HOW 'BOUT A SCALAR
001046  104411                          RANERR                  ;HIGHER ORDER ARRAYS ARE BAD
                                ;
                                ;FIND RESULT WITH SCALAR LEFT ARG
                                ;
001050  013702  003164'         2$:     MOV     LARGPT,R2       ;POINT TO LEFT ARG
001054                                  CALL    @LARGCV         ;CONVERT TO BOOLEAN
001060  105737  003200'                 TSTB    LARGVA          ;WHAT IS IT?
001064  001402                          BEQ     3$              ;YES, EMPTY RESULT
001066  000137  000000G                 JMP     SAME            ;NO, SAME AS RIGHT ARG
001072  000137  000000G         3$:     JMP     EMPTY           ;MAKE EMPTY ARRAY
                                ;
                                ;COMPRESS WITH VECTOR ON LEFT
                                ;
001076  005764  000004          5$:     TST     A.ALEN(R4)      ;IS RIGHT SIDE EMPTY?
001102  001773                          BEQ     3$              ;YES, RESULT IS EMPTY THEN
001104  026327  000004  000001          CMP     A.ALEN(R3),#1   ;Is the left argument a singlton vector?
001112  001756                          BEQ     2$              ;Yes, Then handle like a scalar
001114  026312  000004                  CMP     A.ALEN(R3),(R2) ;IS LEN OF LEFT SAME AS DIM LEN OF RIGHT?
001120  001404                          BEQ     6$              ;YES!
001122  105764  000002                  TSTB    A.DIMS(R4)      ;;SCALAR?
001126  001401                          BEQ     6$              ;;YES, ALL OK
001130  104412                          LENERR                  ;NO, LENGTH ERROR
001132                          6$:     CALL    NBRTRU          ;FIND THE # OF 1'S
001136  005701                          TST     R1              ;ARE THERE ANY '1'S?
001140  001743                          BEQ     2$              ;NO, COMPRESS TO BLANK ARRAY
001142  020163  000004                  CMP     R1,A.ALEN(R3)   ;ARE THERE ANY '0'S?
001146  001003                          BNE     9$              ;YES, RESULT ISN'T THE SAME AS RIGHT ARG
001150  105764  000002                  TSTB    A.DIMS(R4)      ;;IS THE RIGHT ARG A SCALAR?
001154  001335                          BNE     2$              ;;NO, SO SHORTCUT
001156  116446  000002          9$:     MOVB    A.DIMS(R4),-(SP);;SAVE THE DIM COUNT
001162                                  CALL    RESBLK          ;CREATE RESULT BLOCK
                                ;
                                ;DO ACTUAL COMPRESSION
```

MIXED FUNCTION COMPRESSION

```
1759                                        ;
1760 001166  105716                         TSTB    @SP             ;;IS IT A SCALAR ?
1761 001170  001010                         BNE     7$              ;;NO...
1762 001172  013716  005134'                MOV     SKPSIZ,@SP      ;;OH YES IT IS
1763 001176  016346  000006                 MOV     A.DIM1(R3),-(SP);;SAVE # OF ELEMENTS
1764 001202  112764  000001  000002        MOVB    #1,A.DIMS(R4)   ;RESULT IS A VECTOR
1765 001210  000403                         BR      8$              ;;AND CONTINUE
1766 001212  005016                 7$:     CLR     @SP             ;;FOR NON-SCALARS
1767 001214  017746  005122'                MOV     @COMPTR,-(SP)   ;;SAVE # OF ELEMENTS
1768 001220  013704  003206'         8$:     MOV     RARGPT,R4       ;POINT TO FIRST ELEMENT OF RIGHT ARG
1769 001224  011603                 21$:    MOV     @SP,R3          ;GET # OF ELEMENTS TO COMPRESS ALONG DIM
1770 001226  013702  003164'                MOV     LARGPT,R2       ;POINT TO LEFT ARG ELEMENTS
1771 001232  013701  005134'         20$:    MOV     SKPSIZ,R1       ;GET 'SIZE OF GROUPS'
1772 001236                                 CALL    @LARGCV         ;GET THE ELEMENT
1773 001242  105737  003200'                TSTB    LARGVA          ;WELL, IS IT TRUE OR NOT?
1774 001246  001404                         BEQ     22$             ;IF FALSE, DON'T COPY
1775 001250  112420                 17$:    MOVB    (R4)+,(R0)+     ;COPY DATA
1776 001252                                 SOB     R1,17$          ;SEE IF ANY MORE TO COPY
1777 001256  000401                         BR      18$             ;GO TEST NEXT LEFT ELEMENT
1778
1779 001260  060104                 22$:    ADD     R1,R4           ;SKIP ONE GROUP
1780
1781 001262  166604  000002         18$:    SUB     2(SP),R4        ;;CHEAT FOR SCALAR
1782 001266                                 SOB     R3,20$          ;SEE IF MORE IN LEFT ARG
1783 001272  005337  005126'                DEC     COMCNT          ;ARE WE DONE WITH THE WHOLE THING?
1784 001276  001352                         BNE     21$             ;NO, SO DON'T STAND THERE, DO SOME MORE!!
1785 001300  022626                         CMP     (SP)+,(SP)+     ;;GET RID OF BACK-UP FACTOR
1786 001302                                 RETURN                  ;YEA!! RAH!! BRAVO!! WE IS DONE!!
1787
1789 001304                                 LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 55
MIXED FUNCTION          COMPRESSION

   1791 001304                          DOPINS  1EXP,130
   1799 001304                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 001304                          DOPINS  EXPD,62,T.DIM
   1864 001304                                  LIST
```

MIXED FUNCTION COMPRESSION

```
1866 001304                          DOPINS  DROP,104
1907 001304                                  LIST
1908
1909
1910
1911 001304                          DOPINS  TAKE,114
2149 001304                                  LIST
```

MIXED FUNCTION   COMPRESSION

```
   2151 001304                          MOPINS  1REV,146
   2160 001304                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 001304                          MOPINS  REVS,144,T.DIM
   2204 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 58
MIXED FUNCTION          COMPRESSION

   2206 001304                          DOPINS  1ROT,146
   2214 001304                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 001304                          DOPINS  ROT,144,T.DIM
   2328 001304                                  LIST
```

```
   2330 001304                          MOPINS  TRAN,126
   2415 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 60
MIXED FUNCTION          COMPRESSION

   2417 001304                          DOPINS  TRAN,126
   2654 001304                                  LIST
```

```
   2656 001304                          DOPINS  ENCD,76
   2856 001304                                  LIST
```

```
   2858 001304                          DOPINS  DECD,64
   2988 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 63
MIXED FUNCTION          COMPRESSION

 2990 001304                          MOPINS  GDWN,136,T.DIM
 3001 001304                                  LIST
 3002
 3003
 3004 001304                          MOPINS  GDUP,140,T.DIM
 3219 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 64
MIXED FUNCTION          COMPRESSION

   3221 001304                          MOPINS  MMD,122
   3316 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 65
MIXED FUNCTION          COMPRESSION

   3318 001304                          DOPINS  DMD,122
```

```
   3864 001304                                  LIST
```

```
   3866 001304                          DOPINS  MEM,70
   3921 001304                                  LIST
```

```
   3923 001304                          MOPINS  NUB,106
   3986 001304                                  LIST
   3987
   3988
   3989
   3990
   3991
   3992 001304                          DOPINS  UNON,106
   4022 001304                                  LIST
   4023
   4024
   4025
   4026
   4027
   4028 001304                          DOPINS  INTR,66
   4042 001304                                  LIST
   4043
   4044
   4045
   4046
   4047
   4048 001304                          DOPINS  SDIF,50
   4161 001304                                  LIST
```

```
  4163 001304                            DOPINS  SCON,234
  4179 001304                                    LIST
  4180
  4181
  4182 001304                            DOPINS  CONT,112
  4200 001304                                    LIST
  4201
  4202
  4203 001304                            DOPINS  PSUB,236
  4209 001304                                    LIST
  4210
  4211
  4212 001304                            DOPINS  SUBS,116
  4288 001304                                    LIST
```

```
APLOO1  MACRO M1113  08-FEB-80 09:10  PAGE 70
MIXED FUNCTION          COMPRESSION

   4290 001304                              MOPINS  EXE,70
   4336 001304                                      LIST
```

```
   4338 001304                          MOPINS  ENCD,76
   4411 001304                                  LIST
```

```
   4413 001304                          MOPINS  FMT,160
   4423 001304                                  LIST
   4424
   4425
   4426
   4427
   4428
   4429 001304                          DOPINS  FMT,160
```

4953 001304 LIST

```
 4955 001304                        MOPINS  IBM,132,,,MONIBM
 4958 001304                                LIST
 4959
 4960
 4961 001304                        DOPINS  IBM,132
 5046 001304                                LIST
```

```
5048 001304                         MOPINS  QUAD,74,T.NIL+T.SPOP
5075 001304                                 LIST
5076
5077
5078
5079
5080
5081 001304                         DOPINS  QUAD,74,T.NIL+T.SPOP
5097 001304                                 LIST
5098
5099
5100
5101
5102
5103 001304                         MOPINS  QQAD,142,T.NIL+T.SPOP
5135 001304                                 LIST
5136
5137
5138
5139
5140
5141 001304                         DOPINS  QQAD,142,T.NIL+T.SPOP
5152 001304                                 LIST
5153
5154
5155
5156
5157
5158 001304                         MOPINS  DQAD,134,T.NIL+T.SPOP
5192 001304                                 LIST
5193
5194
5195
5196 001304                         DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 001304                                 LIST
```

```
5201 001304                          MOPINS  BXAV,162,T.NIL
5219 001304                                  LIST
5220
5221
5222
5223
5224
5225 001304                          MOPINS  BXCT,164,T.NIL+T.SPOP
5234 001304                                  LIST
5235
5236
5237 001304                          DOPINS  BXCT,164,T.NIL+T.SPOP
5272 001304                                  LIST
5273
5274
5275
5276
5277
5278 001304                          MOPINS  BXIO,166,T.NIL+T.SPOP
5287 001304                                  LIST
5288
5289
5290 001304                          DOPINS  BXIO,166,T.NIL+T.SPOP
5302 001304                                  LIST
5303
5304
5305
5306
5307 001304                          MOPINS  BXLC,212,T.NIL
5340 001304                                  LIST
5341
5342
5343
5344
5345
5346
5347 001304                          MOPINS  BXPP,170,T.NIL+T.SPOP
5356 001304                                  LIST
5357
5358
5359 001304                          DOPINS  BXPP,170,T.NIL+T.SPOP
5381 001304                                  LIST
5382
5383
5384
5385
5386
5387
5388 001304                          MOPINS  BXPW,172,T.NIL+T.SPOP
5397 001304                                  LIST
5398
5399
5400 001304                          DOPINS  BXPW,172,T.NIL+T.SPOP
5417 001304                                  LIST
5418
5419
5420
```

MIXED FUNCTION COMPRESSION

```
 5421
 5422
 5423 001304                        MOPINS  BXRL,174,T.NIL+T.SPOP
 5432 001304                                LIST
 5433
 5434
 5435 001304                        DOPINS  BXRL,174,T.NIL+T.SPOP
 5447 001304                                LIST
 5448
 5449
 5450
 5451
 5452
 5453 001304                        MOPINS  BXTS,200,T.NIL
 5562 001304                                LIST
 5563
 5564
 5565
 5566
 5567
 5568
 5569 001304                        MOPINS  BXWA,176,T.NIL
 5592 001304                                LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 77
MIXED FUNCTION          COMPRESSION

   5594 001304                          MOPINS  BXCR,206
   5716 001304                                  LIST
```

MIXED FUNCTION COMPRESSION

```
 5718 001304                          MOPINS  BXEX,202
 5775 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 79
MIXED FUNCTION          COMPRESSION

   5777 001304                          MOPINS  BXFX,210
   5879 001304                                  LIST
```

```
APL001  MACRO M1113  08-FEB-80 09:10  PAGE 80
MIXED FUNCTION          COMPRESSION

   5881 001304                          MOPINS  BXNC,204
   5950 001304                                  LIST
```

```
   5952 001304                          MOPINS  BXNL,214
   5963 001304                                  LIST
   5964
   5965
   5966 001304                          DOPINS  BXNL,214
   6104 001304                                  LIST
```

```
  6106 001304                            DOPINS  FGET,150,T.DIM
  6330 001304                                    LIST
  6331
  6332
  6333
  6334
  6335
  6336 001304                            DOPINS  FPUT,152,T.DIM
  6485 001304                                    LIST
```

```
   6487 001304                          DOPINS  FPNT,154,T.DIM
   6714 001304                                  LIST
```

```
 6716 001304                        MOPINS  1SCN,130
 6723 001304                                LIST
 6724
 6725
 6726 001304                        MOPINS  SCAN,62,T.DIM
 6732 001304                                LIST
 6733
 6734
 6735 001304                        MOPINS  1RDT,124
 6741 001304                                LIST
 6742
 6743
 6744 001304                        MOPINS  RDCT,60,T.DIM
 7044 001304                                LIST
```

```
   7046 001304                          MOPINS  GETIDN,0
   7193 001304                                  LIST
```

```
7195 001304                             MOPINS  SATAN,0
7196                                    .SBTTL  SATAN                   GENERAL PURPOSE DATA MOVER
7197
7198                                    ;++
7199                                    ;'SATAN' is a general purpose data mover used by Reduction, Scan, and
7200                                    ;Monadic transpose.  It is recursive and is driven by special frames
7201                                    ;on the SP stack.  The format of each upper level frame is shown below:
7202                                    ;(High to Low core)
7203                                    ;
7204                                    ;       RESULT POINTER for this level (Initially zero)
7205                                    ;       RESULT POINTER OFFSET for the next element on this level
7206                                    ;       SOURCE POINTER for this level (Initially zero)
7207                                    ;       COUNT LIMIT of elements along this level (dimension)
7208                                    ;       CURRENT COUNT for this level (Initially zero)
7209                                    ;       SOURCE POINTER OFFSET for the next element on this level
7210                                    ;
7211                                    ;The zero level frame is setup somewhat differently:
7212                                    ;
7213                                    ;       RESULT STORE ROUTINE for Relational Scans
7214                                    ;       RESULT POINTER INCREMENT
7215                                    ;       COUNT LIMIT for Non-associative Scans
7216                                    ;       COUNT limit for this level
7217                                    ;       COUNT INCREMENT for COUNT (Reduction, Assoc Scan:0, Non-assoc Scan:1)
7218                                    ;       SOURCE POINTER INCREMENT (Left argument)
7219                                    ;
7220                                    ;
7221                                    ;'DEVIL' contains the step by which to increment the result pointer.
7222                                    ; (Reduction, Non-associative Scan: 0, Associative Scan: result size in bytes)
7223                                    ;'BLZBUB' points to an optional scalar function routine
7224                                    ;(See Reduction/Scan.)
7225                                    ;
7226                                    ;REGISTER ASSIGNMENTS:
7227                                    ;
7228                                    ; R0 -- Current level count (First level=0)
7229                                    ; R1 -- Result pointer
7230                                    ; R2 -- Input pointer
7231                                    ; R3 -- Temporary pointer
7232                                    ; R4 -- Pointer to SP Stack frames
7233                                    ;
7234                                    ;INITIAL REGISTERS:
7235                                    ;
7236                                    ; R0 -- Number of SP Stack frames
7237                                    ; R1 -- Pointer to first Result location
7238                                    ; R2 -- Pointer to first Argument element
7239                                    ; R4 -- Pointer to frame deepest (Highest in core) in SP Stack
7240                                    ;
7241                                    ;On exit R4 should replace the SP to Pop off the stack frames
7242                                    ;--
7243 001304                                     .PSECT  CODE
7244
7245 001304                             SATAN::
7246 001304  010237  005100'                    MOV     R2,OPTMP5       ;Save source pointer in case of integer overflow
7247 001310  013737  005114' 003202'            MOV     DEVIL,RARGFL    ;Increment for right argument pointer
7248 001316  016637  000014  005072'            MOV     14(SP),OPTMP2   ;Get the new result store rtn ptr just in case
7249
7250 001324                             SATAN1:
7251 001324  005700                             TST     R0              ;At level zero?
```

```
7252 001326 001427                      BEQ     2$              ;Yes, Then do operation
7253 001330 010144                      MOV     R1,-(R4)        ;Save the Result pointer
7254 001332 005744                      TST     -(R4)           ;Skip over the result offset
7255 001334 010244                      MOV     R2,-(R4)        ;Save the source pointer
7256 001336 014444                      MOV     -(R4),-(R4)     ;Reset the count for this level
7257 001340 005744                      TST     -(R4)           ;Skip over the source offset
7258 001342 005300              1$:     DEC     R0              ;Drop down to the next lower level
7259 001344                             CALL    SATAN1          ;And Recurse
7260 001350 062704 000014               ADD     #14,R4          ;Back up one frame
7261 001354 014401                      MOV     -(R4),R1        ;Reset the Result pointer
7262 001356 064401                      ADD     -(R4),R1        ;Increment the result pointer by offset
7263 001360 014402                      MOV     -(R4),R2        ;Reset the Source pointer
7264 001362 024444                      CMP     -(R4),-(R4)     ;Skip over the Count-limit and Count
7265 001364 064402                      ADD     -(R4),R2        ;Increment the source pointer by offset
7266 001366 005364 000002               DEC     2(R4)           ;Count pass for this level
7267 001372 001526                      BEQ     10$             ;All done on this level?
7268 001374 010164 000012               MOV     R1,12(R4)       ;Save the Result pointer
7269 001400 010264 000006               MOV     R2,6(R4)        ;Save Source pointer
7270 001404 000756                      BR      1$              ;And drop down another level
7271                                    ;
7272                                    ;Process the elements on level Zero
7273                                    ;
7274 001406 005744              2$:     TST     -(R4)           ;Skip the result pointer
7275 001410 014437 005102'              MOV     -(R4),OPTMP6    ;Get the Result pointer increment
7276 001414 005744                      TST     -(R4)           ;Skip over the Source pointer
7277 001416 014400                      MOV     -(R4),R0        ;Get the Count
7278 001420 003001                      BGT     3$              ;Is the Count at least one?
7279 001422 104402                      SYSERR                  ;No, SATAN will loop indefinitely
7280 001424 005744              3$:     TST     -(R4)           ;Skip over the count limit increment
7281 001426 014437 003160'              MOV     -(R4),LARGFL    ;Set Left Arg increment
7282 001432 010103                      MOV     R1,R3           ;Save pointer to Result/Right argument
7283 001434                             PUSH    R4              ;Save the frame pointer
7284 001436 013704 005116'              MOV     BLZBUB,R4       ;Get the function number
7285
7286 001442                             CALL    @LARGCV         ;Get first source argument
7287 001446 020027 000001               CMP     R0,#1           ;Is there just one element?
7288 001452 001413                      BEQ     4$              ;Yes, Then just copy it
7289 001454 005737 005114'              TST     DEVIL           ;Are we doing an associative scan?
7290 001460 001010                      BNE     4$              ;Yes, Remember to copy the first element
7291 001462 005300                      DEC     R0              ;Count the first fetch
7292 001464 013737 003200' 003222'      MOV     LARGVA,RARGVA   ;Copy the Non-FPP value to the right argument
7294 001472                             STF     AC1,AC2         ;Copy the FPP value too
7299 001500 000423                      BR      6$              ;And
7300
7301 001502 013737 003200' 003154' 4$:  MOV     LARGVA,RESVAL   ;Copy the argument
7303 001510                             STF     AC1,AC0         ;Copy the FPP value too
7308 001516                             CALL    @RESCVT         ;Store the value into the result
7309 001522 104402                      SYSERR                  ;Can't integer overflow on a copy
7310 001524                             PUSH    RESCVT          ;Save initial result store routine pointer
7311 001530 013737 005072' 003152'      MOV     OPTMP2,RESCVT   ;Get new result store routine pointer
7312 001536                             POP     OPTMP2          ;Now the routine pointers are swapped
7313 001542 000416                      BR      7$              ;All done on this level
7314
7315 001544                     5$:     CALL    @RARGCV         ;Fetch the Previous Result
7316 001550                     6$:     CALL    @LARGCV         ;Fetch the New Argument
7317 001554 010301                      MOV     R3,R1           ;Copy the Pointer to Result
7318 001556                             CALL    @DYDTAB(R4)     ;Call the operator routine
```

```
7319 001562                                    CALL    @RESCVT         ;Store the Result
7320 001566 000434                             BR      INTOVR          ;Integer overflow, Try Floating point
7321 001570 163701 003150'                     SUB     RESLEN,R1       ;Back up result pointer
7322 001574 063701 005102'                     ADD     OPTMP6,R1       ;Increment result pointer
7323 001600                            7$:     SOB     R0,5$           ;Process all elements at level Zero
7324 001604                                    POP     R4              ;Restore the frame pointer
7325
7326 001606 005764 000002              8$:     TST     2(R4)           ;Are we doing non-associative scan?
7327 001612 001416                             BEQ     10$             ;No
7328 001614 026464 000006 000004               CMP     6(R4),4(R4)     ;Yes, End of lowest level loop?
7329 001622 003007                             BGT     9$              ;No
7330 001624 012764 000001 000004               MOV     #1,4(R4)        ;Yes, Reset the count limit
7331 001632 013737 005072' 003152'             MOV     OPTMP2,RESCVT   ;Restore the initial result store routine ptr
7332 001640 000403                             BR      10$
7333 001642 066464 000002 000004       9$:     ADD     2(R4),4(R4)     ;Increment count limit for scan
7334 001650 005200                     10$:    INC     R0              ;Bump up one level
7335 001652 062704 000014                      ADD     #14,R4          ;Point to the next frame
7336 001656                                    RETURN                  ;And return to the calling level
7337                                   ;
7338                                   ;This routine handles integer overflow for the SATAN routine
7339                                   ;
7340
7341 001660 005704                     INTOVR: TST     R4              ;Is the function number negative?
7342 001662 002405                             BLT     2$              ;Yes, Then don't change it
7343 001664 005404                             NEG     R4              ;No, Make it negative
7344 001666 005764 000000G                     TST     DYDTAB(R4)      ;Is there a floating form of the function?
7345 001672 001001                             BNE     2$              ;Yes
7346 001674 104402                     1$:     SYSERR                  ;No, SYSTEM Error
7347 001676 010437 005116'             2$:     MOV     R4,BLZBUB       ;Save the function number
7348 001702 023727 003144' 000004              CMP     RESTYP,#.INTEGER  ;Is the result type integer?
7349 001710 001371                             BNE     1$              ;No, Then die
7350 001712 013701 003124'                     MOV     MONTYP,R1       ;Yes, Get the source argument type
7351 001716 016137 000000G 003176'             MOV     FLARGC(R1),LARGCV ;Setup the new left argument fetch routine
7352 001724 012701 000006                      MOV     #.FLOAT,R1      ;Result type is now floating point
7353 001730 016137 000000G 003152'             MOV     FRESCT(R1),RESCVT ;Setup the new result store routine
7354 001736 010103                             MOV     R1,R3           ;Copy the result type
7355 001740                                    MUL     #5,R3           ;Type times 5
7356 001754 016337 000000G 003220'             MOV     RARGCT(R3),RARGCV ;Setup the right argument fetch routine
7357 001762 013704 000052'                     MOV     PRECSN,R4       ;Get shift count for floating point numbers
7358 001766 005304                             DEC     R4              ;Minus one to convert integer to floating
7359 001770 013702 005114'                     MOV     DEVIL,R2        ;Get result pointer increment
7360 001774                                    ASH     R4,R2           ;Convert to floating type in bytes
7361 002006 010237 005114'                     MOV     R2,DEVIL        ;Yes, Fixup the increment amount
7362 002012 013737 000062' 003150'             MOV     NUMLEN,RESLEN   ;Correct the result length too
7363 002020 013702 003140'                     MOV     RESPTR,R2       ;Save pointer to old result data
7364 002024 013700 003136'                     MOV     RESULT,R0       ;Get pointer to old result block
7365 002030                                    CALL    GETSM2          ;Get a new result block
7366 002034                                    PUSH    <-(SI),-(SI),-(SI)>  ;Save new block's SI frame
7367 002042 010037 003136'                     MOV     R0,RESULT       ;Save the result block pointer
7368 002046 010137 003140'                     MOV     R1,RESPTR       ;Save the result data pointer
7369 002052                                    CALL    KILL            ;Kill, Kill, Kill the old result block
7370 002056                                    POP     <(SI)+,(SI)+,(SI)+>  ;Restore new block's SI frame
7371 002064 013701 005076'                     MOV     OPTMP4,R1       ;Back up the stack
7372 002070 013700 005074'                     MOV     OPTMP3,R0       ;Get the number of frames to process
7373 002074 005300                             DEC     R0              ;Into origin zero
7374 002076 010003                             MOV     R0,R3           ;Copy the number of frames to process
7375
```

```
7376 002100  002675                      BLT     1$              ;Reduct/Scan should have at least one frame
7377 002102  001416                      BEQ     7$              ;Only one frame skip general fixup
7378
7379 002104  005041              6$:     CLR     -(R1)           ;No Result pointer initially
7380 002106  014102                      MOV     -(R1),R2        ;Get the old result pointer offset
7381 002110                              ASH     R4,R2           ;Convert it to floating size
7382 002122  010211                      MOV     R2,(R1)         ;Stow it away
7383 002124  005041                      CLR     -(R1)           ;Source pointer is initially zero
7384 002126  005741                      TST     -(R1)           ;Leave the Count limit alone
7385 002130  005041                      CLR     -(R1)           ;Count is initially zero
7386 002132  005741                      TST     -(R1)           ;Source pointer offset is the same as before
7387 002134                              SOB     R3,6$           ;Do all the frames, but last
7388
7389 002140  013741  003152'     7$:     MOV     RESCVT,-(R1)    ;Get new secondary result store routine ptr
7390 002144  014102                      MOV     -(R1),R2        ;Get the old result pointer increment
7391 002146                              ASH     R4,R2           ;Convert it to floating size
7392 002160  010211                      MOV     R2,(R1)         ;Store the new result pointer increment
7393 002162  024141                      CMP     -(R1),-(R1)     ;Skip over the Count-limit and Count
7394 002164  005761  177776              TST     -2(R1)          ;Doing Non-associative scan?
7395 002170  001402                      BEQ     8$              ;No, Skip the count
7396 002172  012711  000001              MOV     #1,(R1)         ;Yes, Set the count to one
7397 002176  162701  000006      8$:     SUB     #6,R1           ;Skip over the Count increment, Source pointer
7398 002202  010106                      MOV     R1,SP           ;Skip over the return address
7399 002204  013704  005076'             MOV     OPTMP4,R4       ;Point to the stack frames
7400 002210  013701  003140'             MOV     RESPTR,R1       ;Point to first result location
7401 002214  013702  005100'             MOV     OPTMP5,R2       ;Point to first source argument element
7402 002220  000137  001304'             JMP     SATAN           ;Start over again
7404 002224                              LIST
```

```
APLJ01  MACRO M1113  08-FEB-80 09:10  PAGE 87
SATAN                    GENERAL PURPOSE DATA MOVER

   7406 002224                          MOPINS  OUTR,156
   7476 002224                                  LIST
```

```
 7478 002224                    MOPINS  DOUTER,0
 7504 002224                            LIST
 7505
 7506
 7507
 7508
 7509
 7510 002224                    MOPINS  PRDSET,0
 7564 002224                            LIST
 7565
 7566
 7567
 7568
 7569
 7570 002224                    MOPINS  INSPAC,0
 7612 002224                            LIST
```

```
   7614 002224                          DOPINS  INNR,156
```

7960 002224 LIST

```
                                        MOPINS  SUBR,0
                                        .SBTTL  SCIENTIFIC SUBROUTINES
                                        .SBTTL  SUBROUTINE              CLOSE0: Fuzzily close to zero?
                                        ;+
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       This subroutine determines if its argument is within fuzz of zero.
                                        ;       The Z bit is set or cleared depending on the arguments closeness
                                        ;       to zero.
                                        ;-
                                                .PSECT  CODE

                                        CLOSE0::
                                                ABSF    AC0             ;Force AC0 to positive
                                                SUBF    FUZZ,AC0        ;Take off Fuzz
                                                CFCC                    ;Is it close to zero?
                                                BGT     1$              ;No
                                                SEZ                     ;Yes
                                        1$:     RETURN

                                        .SBTTL  SUBROUTINE              .COS:   COSINE
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       WILL RETURN THE COSINE OF THE NUMBER IN AC0, IN AC0.  THE OLD
                                        ;       VALUE WILL BE DESTROYED.  ALL OTHER REGISTERS RESERVED.  THIS ROUTINE
                                        ;       IS FAIRLY ACCRUATE (17 DIGITS) AND CAN BE USED WITH DOUBLE PRECISION.
                                        ;--
                                                .ENABL LSB
                                        .COS::
                                                ABSF    AC0
                                                MOV     #1,-(SP)
                                                BR      1$



                                        .SBTTL  SUBROUTINE              .SIN:   SINE
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       ALMOST THE SAME CODE AS THE COSINE ROUTINE AND HAS THE
                                        ;       SAME SPECS, TOO, EXCEPT DOES THE SINE (NATURALLY)
                                        ;--

                                        .SIN::
                                                CLR     -(SP)
                                        1$:     STFPS   -(SP)
                                                LDFPS   #200
                                                BIT     #200,(SP)       ;Previous mode Single?
                                                BNE     30$             ;Nope
                                                LDCFD   AC0,AC0         ;CLEAR OUT LOW 32 GARBAGE BITS
                                        30$:    STF     AC1,-(SP)
                                                STF     AC2,-(SP)
                                                MOV     R0,-(SP)

                                        ;Quadrant reductions --- ARG = (2/J)X
                                        ;       -1 < ARG < 1
```

```
8019
8020 002342                                        STF     AC0,-(SP)
8021 002350                                        ABSF    AC0
8022 002356                                        MODF    SIN1,AC0
8023 002366                                        MODF    #^F.25,AC1
8024 002376                                        MULF    #^F4,AC1
8025 002406                                        STCFI   AC1,R0
8026 002414  066600  000034                        ADD     34(SP),R0
8027 002420                                        LDF     #^F1,AC1
8028 002430  005200                                INC     R0
8029 002432  006000                                ROR     R0
8030 002434  103404                                BCS     2$
8031 002436                                        SUBF    #^F1,AC0
8032
8033 002446  006000                          2$:   ROR     R0
8034 002450  103003                                BCC     3$
8035 002452                                        NEGF    AC0
8036
8037 002460                                  3$:   TSTF    (SP)+
8038 002466                                        CFCC
8039 002472  100003                                BPL     6$
8040 002474                                        NEGF    AC0
8041
8042 002502                                  6$:   STF     AC0,AC1
8043 002510                                        MULF    AC0,AC1
8044 002516  012700  000046'                       MOV     #SIN2,R0
8045
8046 002522                                        LDF     (R0)+,AC2
8047 002530                                        MULF    AC1,AC2
8048 002536                                        ADDF    (R0)+,AC2
8049 002544                                        MULF    AC1,AC2
8050 002552                                        ADDF    (R0)+,AC2
8051 002560                                        MULF    AC1,AC2
8052 002566                                        ADDF    (R0)+,AC2
8053 002574                                        MULF    AC1,AC2
8054 002602                                        ADDF    (R0)+,AC2
8055 002610                                        MULF    AC2,AC0
8056
8057 002616                                        STF     AC1,AC2
8058 002624                                        ADDF    (R0)+,AC2
8059 002632                                        MULF    AC1,AC2
8060 002640                                        ADDF    (R0)+,AC2
8061 002646                                        MULF    AC1,AC2
8062 002654                                        ADDF    (R0)+,AC2
8063 002662                                        MULF    AC1,AC2
8064 002670                                        ADDF    (R0)+,AC2
8065 002676                                        DIVF    AC2,AC0
8066
8067 002704  012600                                MOV     (SP)+,R0
8068 002706                                        LDF     (SP)+,AC2
8069 002714                                        LDF     (SP)+,AC1
8070 002722                                        LDFPS   (SP)+
8071 002730  005726                                TST     (SP)+
8072 002732                                        RETURN
8073
8074                                 .DSABL LSB
8075
```

```
                                                        .PSECT  TABLES
000036                                                  .WORD   040042, 174603, 067116, 042025
000036  040042  174603  067116  SIN1:
000044  042025
                                                        .WORD   042021, 174005, 170441, 175607
000046  042021  174005  170441  SIN2:
000054  175607
                                                        .WORD   143530, 056427, 005061, 125021
000056  143530  056427  005061
000064  125021
                                                        .WORD   044726, 162341, 133224, 052302
000066  044726  162341  133224
000074  052302
                                                        .WORD   145626, 154170, 031651, 104637
000076  145626  154170  031651
000104  104637
                                                        .WORD   046117, 031130, 175220, 165273
000106  046117  031130  175220
000114  165273
                                                        .WORD   042004, 123513, 026637, 160477
000116  042004  123513  026637
000124  160477
                                                        .WORD   043423, 156142, 064161, 007314
000126  043423  156142  064161
000134  007314
                                                        .WORD   044707, 047147, 032436, 120046
000136  044707  047147  032436
000144  120046
                                                        .WORD   046003, 163716, 123445, 167144
000146  046003  163716  123445
000154  167144


002734                                                  .PSECT  CODE


                                        .SBTTL  SUBROUTINE              .TAN:   TANGENT
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       FINDS THE TANGENT OF AC0 AND RETURNS THE RESULTS IN AC0
                                        ;--

002734                                  .TAN::
002734                                          STF     AC1,-(SP)       ;SAVE AC1
002742                                          STF     AC0,-(SP)       ;SAVE THE ARG TO TAKE TAN OF
002750                                          CALL    .COS            ;GET COSINE OF ARG
002754                                          STF     AC0,AC1         ;AND SAVE IT
002762                                          LDF     (SP)+,AC0       ;GET ARG BACK AGAIN
002770                                          CALL    .SIN            ;FIND THE SINE OF IT
002774                                          DIVF    AC1,AC0         ;TAN(X)=SIN(X)/COS(X)
003002                                          LDF     (SP)+,AC1       ;RESTORE AC1
003010                                          RETURN


                                        .SBTTL  SUBROUTINE              .LN:    NATURAL LOGARITHM
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       THIS SUBROUTINE CALCULATES THE LOG OF THE NUMBER IN AC0 AND
                                        ;       RETURNS THE RESULTS IN AC0.  ALL OTHER REGISTERS ARE PREVSERVED.
                                        ;       IF AC0 IS <=0 THEN A DOMAIN ERROR IS GIVEN.
                                        ;--

003012                                          .PSECT  CODE
003012                                  .LN::
003012                                          TSTF    AC0             ;SEE IF IT'S A LEGAL ARG GIVEN
003020                                          CFCC
```

```
8123 003024  003001                    BGT     1$
8124 003026  104417                    DOMERR
8125
8126 003030                      1$:   CMPF    #^F1,AC0        ;SPECIAL CASE CHECK FOR 1
8127 003040                            CFCC
8128 003044  001004                    BNE     4$
8129 003046                            CLRF    AC0
8130 003054                            RETURN
8131
8132 003056                      4$:   PUSH    R0
8133 003060                            STFPS   -(SP)
8134 003066                            LDFPS   #200            ;D-I MODE FPP
8135 003076  032716  000200            BIT     #200,(SP)       ;Was it single Precision?
8136 003102  001003                    BNE     31$             ;Nope
8137 003104                            LDCFD   AC0,AC0         ;CLEAR OUT LOW 32 GARBAGE BITS
8138 003112                      31$:  STF     AC2,-(SP)       ;SAVE SOME AC'S TILL LATER
8139 003120                            STF     AC1,-(SP)
8140
8141 003126                            STEXP   AC0,-(SP)
8142 003134                            LDEXP   #0,AC0
8143 003144  012700  000156'           MOV     #LN1,R0
8144 003150                            CMPF    (R0)+,AC0
8145 003156                            CFCC
8146 003162  002405                    BLT     2$
8147 003164                            LDEXP   #1,AC0
8148 003174  005316                    DEC     (SP)
8149
8150 003176                      2$:   STF     AC0,AC1
8151 003204                            SUBF    #^F1,AC0
8152 003214                            ADDF    #^F1,AC1
8153 003224                            DIVF    AC1,AC0
8154 003232                            STF     AC0,AC1
8155 003240                            MULF    AC0,AC1
8156
8157 003246                            LDF     (R0)+,AC2
8158 003254                            MULF    AC1,AC2
8159 003262                            ADDF    (R0)+,AC2
8160 003270                            MULF    AC1,AC2
8161 003276                            ADDF    (R0)+,AC2
8162 003304                            MULF    AC1,AC2
8163 003312                            ADDF    (R0)+,AC2
8164 003320                            MULF    AC2,AC0
8165
8166 003326                            LDF     AC1,AC2
8167 003334                            ADDF    (R0)+,AC2
8168 003342                            MULF    AC1,AC2
8169 003350                            ADDF    (R0)+,AC2
8170 003356                            MULF    AC1,AC2
8171 003364                            ADDF    (R0)+,AC2
8172
8173 003372                            DIVF    AC2,AC0
8174 003400                            LDCIF   (SP)+,AC1
8175 003406                            MULF    (R0)+,AC1
8176 003414                            ADDF    AC1,AC0
8177
8178 003422                            LDF     (SP)+,AC1       ;RESTORE SOME REGISTERS
8179 003430                            LDF     (SP)+,AC2
```

```
8180 003436                                        LDFPS   (SP)+
8181 003444                                        POP     R0
8182 003446                                        RETURN
8183
8184 000156                                        .PSECT  TABLES
8185
8186 000156  040065  002363  031771  LN1:        .WORD   40065,2363,31771,157145
     000164  157145
8187 000166  037727  114303  110107              .FLT4   .4210673712179797145
     000174  114145
8188 000176  141032  031773  064222              .FLT4   -.9637690933686865932 4E+1
     000204  040261
8189 000206  041367  124211  021611              .FLT4   .3095729282153765006 2264E+2
     000214  114442
8190 000216  141300  016201  002154              .FLT4   -.2401391795592105098 68484E+2
     000224  010216
8191 000226  141016  111747  007541              .FLT4   -.8911109027937831233 7E+1
     000234  052530
8192 000236  041233  154404  136454              .FLT4   .1948096607008897305 1623E+2
     000244  022153
8193 000246  141100  016201  002154              .FLT4   -.1200695897796052547 17525E+2
     000254  010216
8194 000256  040061  071027  173721              .FLT4   .6931471805599453094 17232121458
     000264  147572
8195
8196
8197                                   .SBTTL  SUBROUTINE                .EXP:   EXPONENTIAL
8198                                   ;++
8199                                   ; FUNCTIONAL DESCRIPTION:
8200                                   ;       THIS ROUTINE WILL FIND THE EXPONENTIAL, e^x or EXP(x), OF AC0
8201                                   ;       AND RETURN THE RESULT IN AC0. ALL OTHER REGISTERS ARE PRESERVED.
8202                                   ;       THIS ROUTINE DOES 17 DIGIT CALCLATIONS AND IS CAPABLE OF DOUBLE
8203                                   ;       PRECISION CALCULATIONS.  For further information see "Computer
8204                                   ;       Approximations", John F. Hart, et.al., p100-102,
8205                                   ;       (approximation number 1067, maybe? ).
8206                                   ;--
8207 003450                                        .PSECT  CODE
8208 003450                           .EXP::
8209 003450                                        PUSH    R0              ;Save R0
8210 003452                                        STFPS   -(SP)           ;SAVE CURRENT FPP STATUS
8211 003460                                        LDFPS   #200            ;D-I MODE
8212 003470  032716  000200                        BIT     #200,(SP)       ;Previous mode Single?
8213 003474  001003                                BNE     32$             ;Nope
8214 003476                                        LDCFD   AC0,AC0         ;CLEAR OUT LOW 32 GARBAGE BITS
8215 003504                           32$:         STF     AC2,-(SP)       ;SAVE AC2
8216 003512                                        STF     AC1,-(SP)       ;    AND AC1
8217 003520                                        TSTF    AC0             ;CHECK FOR SPECIAL CASE OF e^0
8218 003526                                        CFCC
8219 003532  001005                                BNE     1$              ;IF NOT, DO IT THE APPROX METHOD
8220
8221 003534                                        LDF     #^F1,AC0        ;e^0 = 1
8222 003544  000562                                BR      8$              ;RESTORE AND RETURN!
8223
8224 003546  012700  000266'          1$:         MOV     #EXP1,R0
8225 003552                                        MODF    (R0)+,AC0       ;EXP(x)=2^(x*LOG2(e))
8226 003560                                        CFCC
8227 003564  100413                                BMI     2$
```

```
8228 003566                          STCFI   AC1,-(SP)      ;STORE INTEGER PART
8229 003574                          CFCC
8230 003600  103516                  BCS     4$             ;The argument is too big
8231 003602                          SUBF    #^F.5,AC0
8232 003612  000413                  BR      3$
8233
8234 003614                  2$:     STCFI   AC1,-(SP)      ;Store the integer part
8235 003622                          CFCC
8236 003626  103517                  BCS     6$             ;The argument is too negative
8237 003630  005316                  DEC     (SP)
8238 003632                          ADDF    #^F.5,AC0
8239
8240 003642                  3$:     STF     AC0,AC1        ;-.5< x <+.5
8241 003650                          MULF    AC1,AC1        ;x**2
8242
8243 003656                          LDF     (R0)+,AC2      ;P2
8244 003664                          MULF    AC1,AC2
8245 003672                          ADDF    (R0)+,AC2      ;P1
8246 003700                          MULF    AC1,AC2
8247 003706                          ADDF    (R0)+,AC2      ;P0
8248 003714                          MULF    AC2,AC0        ;xP(x**2)
8249
8250 003722                          LDF     AC1,AC2
8251 003730                          ADDF    (R0)+,AC2      ;Q1
8252 003736                          MULF    AC1,AC2
8253 003744                          ADDF    (R0)+,AC2      ;Q0
8254
8255 003752                          STF     AC2,AC1        ;Q(x**2)
8256 003760                          SUBF    AC0,AC1        ;Q-xP
8257 003766                          ADDF    AC2,AC0        ;Q+xP
8258 003774                          DIVF    AC1,AC0        ;(Q+xP)/(Q-xP)
8259 004002                          MULF    (R0)+,AC0      ;I don't know what this is! What happened to Q2?
8260
8261 004010                          STEXP   AC0,-(SP)      ;Get the exponent of the fractional part
8262 004016  005766  000002          TST     2(SP)          ;Is the integer exponent positive?
8263 004022  002414                  BLT     5$             ;No
8264
8265 004024  062616                  ADD     (SP)+,(SP)     ;Correct the exponent
8266 004026  102403                  BVS     4$             ;The exponent is way too big, Domain error
8267 004030  021627  000177          CMP     (SP),#177      ;Is the exponent just a little too big?
8268 004034  003422                  BLE     7$             ;No, All is Ok
8269 004036  005726          4$:     TST     (SP)+          ;Clean up the stack
8270 004040                          LDF     EXPBIG,AC0     ;Get the largest possible number
8271 004050  000261                  SEC                    ;Note the overflow
8272 004052  000417                  BR      8$             ;Now return
8273
8274 004054  062616          5$:     ADD     (SP)+,(SP)     ;Correct the exponent
8275 004056  102403                  BVS     6$             ;The result is very very small
8276 004060  021627  177601          CMP     (SP),#-177     ;Is the result just a little small?
8277 004064  002006                  BGE     7$             ;No, Then Ok
8278 004066  005726          6$:     TST     (SP)+          ;Clean up the stack
8279 004070                          CLRF    AC0            ;Use zero for such a tiny result
8280 004076  000261                  SEC                    ;Note the underflow
8281 004100  000404                  BR      8$
8282
8283 004102                  7$:     LDEXP   (SP)+,AC0      ;Attach the exponent to the result
8284 004110  000241                  CLC                    ;SHOW GOODNESS
```

```
8285
8286 004112                          8$:     LDF     (SP)+,AC1       ;RESTORE STUFF ON STACK
8287 004120                                  LDF     (SP)+,AC2
8288 004126                                  LDFPS   (SP)+
8289 004134                                  POP     R0              ;Restore R0
8290 004136                                  RETURN
8291
8292 000266                                  .PSECT  TABLES
8293
8294 000266 040270 125073 024534 EXP1:   .FLT4   1.44269504088896340735992460      ;LOG2(e) = 1/LN(2)
     000274 013761
8295 000276 036675 027102 125560         .FLT4   .230933477537502336240E-1        ;P2
     000304 136652
8296 000306 041241 116724 114237         .FLT4   .20202065651286927227886E+2      ;P1
     000314 060333
8297 000316 042675 036404 077563         .FLT4   .15139067990543389158943280E+4   ;P0
     000324 046675
8298 000326 042151 027450 075350         .FLT4   .23318421142748162379029 5E+3    ;Q1
     000334 112503
8299 000336 043210 100661 076072         .FLT4   .43682116627275584984968 4E+4    ;Q0
     000344 062453
8300 000346 040265 002363 031771         .FLT4   1.41421356237309504880168 7      ;You got me what this one is
     000354 157144
8301 000356 077777 177777 177777 EXPBIG: .WORD   077777,177777,177777,177777     ;Biggest number we can handle
     000364 177777
8302
8303
8304
8305
8306
8307                                 .SBTTL  SUBROUTINE              .ACOS
8308                                 ;++
8309                                 ; FUNCTIONAL DESCRIPTION:
8310                                 ;       FINDS THE ARCCOSINE OF AC0
8311                                 ;--
8312
8313 000366                           .ACOS::
8314 000366                                  CALL    .ASIN           ;ACOS(X) = TT/2 - ASIN(X)
8315 000372                                  STF     AC0,-(SP)       ;SAVE IT
8316 000400                                  LDF     PI,AC0          ;GET PI
8317 000410                                  MULF    #^F0.5,AC0      ;    PI/2
8318 000420                                  SUBF    (SP)+,AC0       ;ACOS(X)
8319 000426                                  RETURN
8320
8321
8322
8323
8324                                 .SBTTL  SUBROUTINE              .ASIN:  ARCSINE
8325                                 ;++
8326                                 ; FUNCTIONAL DESCRIPTION:
8327                                 ;       FINDS THE ARCSINE OF AC0
8328                                 ;--
8329
8330 000430                           .ASIN:
8331 000430                                  STF     AC1,-(SP)       ;ASIN(X)=ATAN(X/SQRT(1-X^2))
8332 000436                                  STF     AC0,AC1
8333 000444                                  ABSF    AC0             ;MAKE -1 -> 1
```

```
8334 000452                            SUBF    #^F1.,AC0       ;CHECK FOR NEAR PLUS OR MINUS ONE
8335 000462                            CALL    CLOSE0          ;IS IT ?
8336 000466  001025                    BNE     2$              ;NO, SKIP AROUND
8337 000470                            LDF     PI,AC0          ;GET PI
8338 000500                            MULF    #^F0.5,AC0      ;    PI/2
8339 000510                            TSTF    AC1             ;POSITIVE OR NEGATIVE ?
8340 000516                            CFCC
8341 000522  002003                    BGE     1$              ;POSITIVE...
8342 000524                            NEGF    AC0             ;NEGATIVE
8343 000532                    1$:     LDF     (SP)+,AC1       ;RESTORE AC1
8344 000540                            RETURN                  ;FINISHED
8345
8346 000542                    2$:     STF     AC1,AC0
8347 000550                            MULF    AC0,AC0
8348 000556                            NEGF    AC0
8349 000564                            ADDF    #^F1,AC0
8350 000574                            CALL    .SQRT
8351 000600                            DIVF    AC0,AC1
8352 000606                            STF     AC1,AC0
8353 000614                            LDF     (SP)+,AC1
8354                           ;        BR      .ATAN
8355
8356
8357
8358
8359                           .SBTTL  SUBROUTINE              .ATAN:  ARCTANGENT
8360                           ;++
8361                           ; FUNCTIONAL DESCRIPTION:
8362                           ;       FINDS THE ARCTANGENT OF AC0
8363                           ;--
8364
8365 000622                    .ATAN::
8366 000622                            STF     AC1,-(SP)       ;SAVE AC1
8367 000630                            STF     AC2,-(SP)       ;AND AC2
8368 000636                            STF     AC3,-(SP)       ;AND AC3
8369 000644                            STFPS   -(SP)           ;AND THE FPP STATUS
8370 000652                            PUSH    R0              ;AND LAST BUT NOT LEAST R0
8371 000654                            LDFPS   #40200          ;SET D-I MODE WITH NO INTERUPTS
8372 000664  032716  000200            BIT     #200,(SP)       ;Previous Mode Single?
8373 000670  001003                    BNE     33$             ;Nope
8374 000672                            LDCFD   AC0,AC0         ;CLEAR OUT LOW 32 GARBAGE BITS
8375
8376 000700                    33$:    TSTF    AC0             ;SEE WHAT SIGN THE NUMBER IS
8377 000706                            CFCC
8378 000712  001414                    BEQ     2$              ;IF ZERO WERE DONE ALREADY
8379 000714  002403                    BLT     1$              ;TAKE CARE OF NEGITIVE NUMBERS
8380 000716                            CALL    3$                      ;GET THE ARCTAN
8381 000722  000410                    BR      2$              ;AND EXIT
8382
8383 000724                    1$:     NEGF    AC0             ;ATN(-X)=-ATN(X)
8384 000732                            CALL    3$                      ;FIND THE ARCTAN
8385 000736                            NEGF    AC0             ;AND REMEMBER TO NEGATE IT
8386
8387 000744                    2$:     POP     R0              ;RESTORE R0
8388 000746                            LDFPS   (SP)+           ;RESTORE THE FPP STATUS
8389 000754                            LDF     (SP)+,AC3       ;RESTORE AC3
8390 000762                            LDF     (SP)+,AC2       ;AND AC2
```

```
8391 000770                                    LDF     (SP)+,AC1       ;AND AC1
8392 000776                                    RETURN                  ;RETURN WITH THE ARCTAN IN AC0
8393
8394                                ;DO SOME WORK WITH SHUFFLING RANGES ON THE ARG
8395
8396 001000  012700  001356'       3$:     MOV     #ATAN1,R0       ;GET POINTER TO CONSTANTS
8397 001004                                CMPF    (R0)+,AC0
8398 001012                                CFCC
8399 001016  002057                        BGE     5$
8400 001020                                CMPF    ATAN3,AC0
8401 001030                                CFCC
8402 001034  003023                        BGT     4$
8403 001036                                LDF     #^F1,AC1
8404 001046                                DIVF    AC0,AC1
8405 001054                                STF     AC1,AC0
8406 001062                                CALL    5$
8407 001066                                NEGF    AC0
8408 001074                                ADDF    (R0)+,AC0
8409 001102                                RETURN
8410
8411 001104                        4$:     STF     AC0,AC1
8412 001112                                SUBF    #^F1,AC0
8413 001122                                ADDF    #^F1,AC1
8414 001132                                DIVF    AC1,AC0
8415 001140                                CALL    5$
8416 001144                                ADDF    ATAN2,AC0
8417 001154                                RETURN
8418
8419                                ;DO     THE ACTUAL SERIES ON THE PROPERLY REDUCED VALUE
8420
8421 001156                        5$:     STF     AC0,AC3
8422 001164                                MULF    AC3,AC3
8423 001172                                LDF     (R0)+,AC1
8424
8425 001200                                MULF    AC3,AC1
8426 001206                                ADDF    (R0)+,AC1
8427 001214                                MULF    AC3,AC1
8428 001222                                ADDF    (R0)+,AC1
8429 001230                                MULF    AC3,AC1
8430 001236                                ADDF    (R0)+,AC1
8431 001244                                MULF    AC3,AC1
8432 001252                                ADDF    (R0)+,AC1
8433
8434 001260                                STF     AC3,AC2
8435 001266                                ADDF    (R0)+,AC2
8436 001274                                MULF    AC3,AC2
8437 001302                                ADDF    (R0)+,AC2
8438 001310                                MULF    AC3,AC2
8439 001316                                ADDF    (R0)+,AC2
8440 001324                                MULF    AC3,AC2
8441 001332                                ADDF    (R0)+,AC2
8442
8443 001340                                DIVF    AC2,AC1
8444 001346                                MULF    AC1,AC0
8445 001354                                RETURN
8446
8447 001356                                .PSECT  TABLES
```

SUBROUTINE .ATAN: ARCTANGENT

```
 8448
 8449
 8450 001356 037724 011714 147747 ATAN1:  .FLT4   .4142135623730950488O
      001364 074622
 8451 001366 037442 145026 075504         .FLT4   .1589740288482307048
      001374 015622
 8452 001376 040725 021566 115517         .FLT4   .6660579017009262658E+1
      001404 050310
 8453 001406 041443 160206 172714         .FLT4   .4096926483210225637E+2
      001414 025442
 8454 001416 041632 172223 076027         .FLT4   .7747768771920420862E+2
      001424 056646
 8455 001426 041462 025125 006152         .FLT4   .4454134005929068032E+2
      001434 126065
 8456 001436 041170 010112 141724         .FLT4   .1550397755142198753E+2
      001444 064326
 8457 001446 041573 053776 025372         .FLT4   .6283593051103237683E+2
      001454 071524
 8458 001456 041670 123114 051576         .FLT4   .9232480107230097484E+2
      001464 075022
 8459 001466 041462 025125 006152         .FLT4   .4454134005929068032E+2
      001474 126065
 8460 001476 040311 007732 121041         .FLT4   1.570796326794896619230
      001504 064302
 8461
 8462 001506 040111 007732 121041 ATAN2:  .WORD   40111,7732,121041,64302
      001514 064302
 8463 001516 040432 101171 114774 ATAN3:  .FLT4   2.414213562373095048800
      001524 167462
```

```
8465                                    .SBTTL  SUBROUTINE              .SINH:  HYPERBOLIC SINE
8466                                    ;++
8467                                    ; FUNCTIONAL DESCRIPTION:
8468                                    ;       FINDS THE HYPERBOLIC SINE OF AC0
8469                                    ;--
8470
8471 001526                            .SINH::
8472 001526                                    CALL    .EXP            ;SINH(X)=(EXP(X)-1/EXP(X))/2
8473 001532  103001                            BCC     1$              ;Overflowed?
8474 001534  104417                            DOMERR                  ;Yes, Domain error
8475 001536                            1$:     STF     AC1,-(SP)
8476 001544                                    LDF     #^F1,AC1
8477 001554                                    DIVF    AC0,AC1
8478 001562                                    SUBF    AC1,AC0
8479 001570                                    DIVF    #^F2,AC0
8480 001600                                    LDF     (SP)+,AC1
8481 001606                                    RETURN
8482
8483
8484
8485
8486                                    .SBTTL  SUBROUTINE              .COSH:  HYPERBOLIC COSINE
8487                                    ;++
8488                                    ; FUNCTIONAL DESCRIPTION:
8489                                    ;       FINDS THE HYPERBOLIC COSINE OF AC0
8490                                    ;--
8491
8492 001610                            .COSH::
8493 001610                                    CALL    .EXP            ;COSH(X)=(EXP(X)+1/EXP(X))/2
8494 001614  103001                            BCC     1$              ;Overflowed?
8495 001616  104417                            DOMERR                  ;Yes, Domain error
8496 001620                            1$:     STF     AC1,-(SP)
8497 001626                                    LDF     #^F1,AC1
8498 001636                                    DIVF    AC0,AC1
8499 001644                                    ADDF    AC1,AC0
8500 001652                                    DIVF    #^F2,AC0
8501 001662                                    LDF     (SP)+,AC1
8502 001670                                    RETURN
8503
8504
8505
8506
8507                                    .SBTTL  SUBROUTINE              .TANH:  HYPERBOLIC TANGENT
8508                                    ;++
8509                                    ; FUNCTIONAL DESCRIPTION:
8510                                    ;       FINDS THE HYPERBOLIC TANGENT OF AC0
8511                                    ;--
8512
8513 001672                            .TANH::
8514 001672                                    STF     AC0,-(SP)       ;Save argument for its sign
8515 001700                                    CALL    .EXP            ;TANH(x)=(EXP(x)-1/EXP(x))/(EXP(x)+1/EXP(x))
8516 001704  103016                            BCC     1$              ;Did EXP(x) overflow?
8517 001706                                    LDF     #^F1,AC0        ;Yes, Then TANH(infinity) = 1
8518 001716                                    TSTF    (SP)            ;What was the argument's sign?
8519 001724                                    CFCC
8520 001730  002043                            BGE     2$              ;Plus
8521 001732                                    NEGF    AC0             ;Minus
```

```
                                        BR      2$              ;All done, return
                                1$:     STF     AC1,-(SP)
                                        STF     AC2,-(SP)
                                        LDF     #^F1,AC1
                                        DIVF    AC0,AC1
                                        STF     AC0,AC2
                                        ADDF    AC1,AC2
                                        SUBF    AC1,AC0
                                        DIVF    AC2,AC0
                                        LDF     (SP)+,AC2
                                        LDF     (SP)+,AC1
                                2$:     TSTF    (SP)+           ;Get rid of saved argument
                                        RETURN



                                .SBTTL  SUBROUTINE              .ATANH: HYPERBOLIC ARCTANGENT
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       FINDS THE HYBERBOLIC ARCTANGENT OF THE NUMBER IN AC0, AND
                                ;       RETURNS THE RESULTS IN AC0.
                                ;--
                                        .PSECT  CODE
                                .ATANH::
                                        STF     AC1,-(SP)       ;SAVE SOME REGISTERS
                                        LDF     #^F1,AC1        ;ATANH(X)=LN((1+X)/(1-X))/2
                                        SUBF    AC0,AC1
                                        ADDF    #^F1,AC0
                                        DIVF    AC1,AC0
                                        CALL    .LN
                                        DIVF    #^F2,AC0
                                        LDF     (SP)+,AC1
                                        RETURN



                                .SBTTL  SUBROUTINE              .ACOSH: HYPERBOLIC ARCCOSINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       FINDS THE HYPERBOLIC ARCCOSINE OF A NUMBER IN AC0, AND
                                ;       RETURNS THE RESULT IN AC0.
                                ;--
                                        .ENABL LSB
                                .ACOSH::
                                        STF     AC0,-(SP)       ;ACOSH(X)=X+LN(SQRT(X^2-1))
                                        MULF    AC0,AC0
                                        SUBF    #^F1,AC0
                                1$:     CALL    .SQRT
                                        ADDF    (SP)+,AC0
                                        CALL    .LN
                                        RETURN
```

```
                                .SBTTL  SUBROUTINE              .ASINH: HYPERBOLIC ARCSINE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       FINDS THE HYPERBOLIC ARCSINE OF THE NUMBER IN AC0, AND
                                ;       RETURNS THE RESULT IN AC0.
                                ;--

004272                          .ASINH::
004272                                  STF     AC0,-(SP)       ;ASINH(X)=X+LN(SQRT(X^2+1))
004300                                  MULF    AC0,AC0
004306                                  ADDF    #^F1,AC0
004316  000755                          BR      1$
                                        .DSABL LSB




                                .SBTTL  SUBROUTINE              .FACT:  GENERALIZED FACTORIAL
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       GENERALIZED FACTORIAL ROUTINE.  FOR NEGATIVE OR FRACTIONAL ARGUMENTS
                                ;       THE GAMMA FUNCTION IS APPLIED AFTER REDUCING THE ARGUMENT TO THE
                                ;       RANGE 0<= X <=1.  COEFFICIENTS OBTAINED FROM HART, ET AL. ''COMPUTER
                                ;       APPROXIMATIONS'' ROUTINE #5243.  PRECISION: 17.02 DIGITS.
                                ;--
004320                                  .PSECT  CODE

004320                          .FACT::
004320                                  PUSH    <R0,R1>         ;SAVE R0 and R1
004324                                  STF     AC3,-(SP)       ;SAVE REGISTERS
004332                                  STF     AC2,-(SP)
004340                                  STF     AC1,-(SP)
004346                                  STFPS   -(SP)           ;SAVE FPP STATUS
004354                                  LDFPS   #200            ;SET DOUBLE MODE
004364  032716  000200                  BIT     #200,(SP)       ;Is it in Single Mode?
004370  001003                          BNE     1$              ;Nope
004372                                  LDCFD   AC0,AC0         ;CONVERT FROM SINGLE TO DOUBLE IF NEEDED
004400                          1$:     STD     AC0,AC2         ;COPY ARGUMENT
004406                                  STD     AC0,AC1
004414                                  LDD     #^F1.0,AC0      ;LOAD COMMON CONSTANT
004424                                  MODD    AC0,AC2         ;CHECK FOR INTEGER
004432                                  TSTD    AC1             ;IS ARGUMENT POSITIVE?
004440                                  CFCC
004444  001577                          BEQ     GAMDON          ;0!=1
004446  100426                          BMI     4$              ;YES
004450                                  CMPD    #^F32.0,AC1     ;POSITIVE ARG TOO BIG?
004460                                  CFCC
004464  003004                          BGT     3$              ;NO, ARG IS IN BOUNDS
004466                          2$:     LDFPS   (SP)+           ;RESTORE FPP STATUS
004474  104417                          DOMERR                  ;AND COMPLAIN ABOUT ARG
                                ;+
                                ;POSITIVE ARG
                                ;-
004476  012701  000001          3$:     MOV     #1,R1           ;SET POSITIVE ARG FLAG
004502                                  LDD     #^F-1.0,AC2     ;INCREMENT IS NEGATIVE
```

```
8636 004512                              STCDI   AC3,R0          ;GET INTEGER PART OF ARG
8637 004520  001443                      BEQ     GAMMA           ;IF ONLY FRACTION, JUST DO GAMMA APPROX.
8638 004522  000432                      BR      5$              ;ELSE FIRST REDUCE ARG DOWN TO GAMMA RANGE
8639                                 ;+
8640                                 ;NEGATIVE ARG
8641                                 ;-
8642 004524                          4$: CMPD    #^F-32.0,AC1    ;IS ARG TOO NEGATIVE?
8643 004534                              CFCC
8644 004540  003352                      BGT     2$              ;YUP, ARG OUT OF BOUNDS
8645 004542  012701  177777              MOV     #-1,R1          ;SET NEGATIVE ARG FLAG
8646 004546                              TSTD    AC2             ;IS ARG A NEGATIVE INTEGER?
8647 004554                              CFCC
8648 004560  001742                      BEQ     2$              ;YES, BAD ARG (CAUSES DIVISION BY ZERO)
8649 004562                              STD     AC0,AC2         ;INCREMENT IS 1
8650 004570                              ADDD    AC0,AC1         ;FUDGE NEGATIVE ARG
8651 004576                              STCDI   AC3,R0          ;GET INTEGER PART
8652 004604  005400                      NEG     R0              ;ELSE MAKE COUNT POSITIVE
8653 004606  005200                      INC     R0              ;PLUS 1 TO INCLUDE FRACTION
8654                                 ;+
8655                                 ;REDUCE ARG TO FRACTION 0<= X <=1.
8656                                 ;-
8657 004610                          5$: MULD    AC1,AC0         ;MULTIPLY RESULT BY ARG
8658 004616                              ADDD    AC2,AC1         ;REDUCE ARG
8659 004624                              SOB     R0,5$           ;REDUCE ARG TO MANAGEABLE SIZE
8660                                 ;+
8661                                 ;PERFORM GAMMA FUNCTION APPROXIMATION ON FRACTIONAL REMAINDER
8662                                 ;-
8663 004630                          GAMMA:  TSTD    AC1             ;WAS ARG AN INTEGER?
8664 004636                              CFCC
8665 004642  001500                      BEQ     GAMDON          ;YES, WE HAVE RESULT (STANDARD FACTORIAL)
8666 004644                              CMPD    #^F1.0,AC1      ;IS RESULT IN BOUNDS?
8667 004654                              CFCC
8668 004660  002007                      BGE     1$              ;YES
8669 004662                              SUBD    #^F1.0,AC1      ;NO, FUDGE FOR POSITIVE FRACTIONAL ARG
8670 004672                              CFCC                    ;DONE?
8671 004676  001462                      BEQ     GAMDON          ;YES!
8672
8673 004700                          1$: PUSH    R1              ;SAVE POSITIVE/NEGATIVE ARG FLAG
8674 004702  012701  002050'             MOV     #GAMTAB,R1      ;POINT TO COEFFICIENT TABLE
8675 004706  012700  000006              MOV     #6.,R0          ;THERE ARE 6 TERMS
8676 004712                              STD     AC1,AC3         ;FIRST COEFF OF DENOMINATOR IS 1
8677 004720                              LDD     (R1)+,AC2       ;GET FIRST COEFF OF NUMERATOR
8678 004726                              ADDD    (R1)+,AC3       ;ADD SECOND COEFF TO DENOM (1 MORE TERM)
8679
8680 004734                          2$: MULD    AC1,AC2         ;MUL NUMERATOR BY ARG
8681 004742                              MULD    AC1,AC3         ;AND DENOMINATOR
8682 004750                              ADDD    (R1)+,AC2       ;ADD NEXT TERM TO NUM.
8683 004756                              ADDD    (R1)+,AC3       ;AND TO DENOM.
8684 004764                              SOB     R0,2$           ;6 TERMS
8685
8686 004770                              DIVD    AC3,AC2         ;MAKE RATIONAL APPROX.
8687 004776  005726                      TST     (SP)+           ;CHECK SIGN OF ARG
8688 005000  100404                      BMI     4$              ;NEGATIVE ARG
8689 005002                              MULD    AC0,AC2         ;MULTIPLY BY ARG REDUCTION VALUE
8690 005010  000403                      BR      5$              ;AND CLEAN UP
8691
8692 005012                          4$: DIVD    AC0,AC2         ;DIVIDE GAMMA BY REDUCTION VALUE
```

```
8693
8694 005020                                5$:     ADDD    #^F1.0,AC1      ;BUMP UP ARGUMENT
8695 005030                                        DIVD    AC1,AC2         ;AND SCALE RESULT
8696 005036                                        STD     AC2,AC0         ;PUT RESULT INTO CORRECT REG
8697
8698 005044                                GAMDON: LDFPS   (SP)+           ;POP FPP STATUS
8699 005052                                        LDF     (SP)+,AC1       ;RESTORE REGS
8700 005060                                        LDF     (SP)+,AC2
8701 005066                                        LDF     (SP)+,AC3
8702 005074                                        POP     <R1,R0>         ;Restore R1 and R0
8703 005100                                        RETURN                  ;THAT WAS HARD WORK!
8704
8705
8706                                        ;+
8707                                        ;TABLE OF POLYNOMIAL COEFFS FOR GAMMA FUNCTION
8708                                        ;-
8709 002050                                        .PSECT  TABLES
8710 002050  140727  153242  163350        GAMTAB: .FLT4   -.6744950724592528991 8E+1      ;P6
     002056  032215
8711 002060  141270  123404  076130                .FLT4   -.2308155152458012456 2E+2      ;Q6
     002066  012650
8712 002070  141510  067515  065111                .FLT4   -.5010869375297095301 6E+2      ;P5
     002076  024263
8713 002100  042075  077614  043022                .FLT4   .1894982341570280164 2E+3       ;Q5
     002106  027572
8714 002110  142333  125113  176657                .FLT4   -.4393304440600256761 4E+3      ;P4
     002116  075737
8715 002120  142371  101646  141245                .FLT4   -.4990285266214390483 4E+3      ;Q4
     002126  011263
8716 002130  142773  010340  074264                .FLT4   -.2008527401307279121 5E+4      ;P3
     002136  152065
8717 002140  142677  011556  144553                .FLT4   -.1528607273779522024 8E+4      ;Q3
     002146  154176
8718 002150  143410  165327  054121                .FLT4   -.8762710297852148956 1E+4      ;P2
     002156  172630
8719 002160  043433  050472  145404                .FLT4   .9940307415082770901 5E+4       ;Q2
     002166  175461
8720 002170  143643  026671  036161                .FLT4   -.2088686178926988736 4E+5      ;P1
     002176  072153
8721 002200  143072  043052  050302                .FLT4   -.2980385330925664993 2E+4      ;Q1
     002206  136746
8722 002210  144045  070660  101665                .FLT4   -.4235368950974408964 7E+5      ;P0
     002216  164443
8723 002220  144045  070660  101665                .FLT4   -.4235368950974409001 0E+5      ;Q0
     002226  164444
8725 002230                                        LIST
```

```
   8727 002230                          MOPINS  RAND,0
   8771 002230                                  LIST
```

```
APLOO1  MACRO M1113  08-FEB-80 09:10  PAGE 94
SUBROUTINE                .FACT:  GENERALIZED FACTORIAL

      1          000001                              .END
```

APL001 MACRO M1113 08-FEB-80 09:10 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHREP  = 000105          CS.BOO= 000010          D$COMP  001012RG  020 D.SUBS= 000006
AC1    =%000001          CHRERR= 104431          CS.FLO= 000020          D$LOG   000362RG  020 D.TAKE= 000002
AC2    =%000002          CHRFIL= 000177          CS.IDN= 000040          D$PWR   000000RG  020 D.TRAN= 000004
AC3    =%000003          CHRFRO= 000135          CS.ONE= 000100          D$1COM  000772RG  020 D.UNON= 000006
AC4    =%000004          CHRLBR= 000073          CTEPS   000152R  025    D.AND = 000000          D.$MAX= 000000
AC5    =%000005          CHRLPR= 000072          CTMAX   000212R  025    D.BXCT= 000007          D.$MIN= 000000
ALLINT  003274R  025     CHRMNS= 000137          CTRMAX  000172R  025    D.BXIO= 000007          D.$MUL= 000000
ARYRET  003276R  025     CHRMUL= 000075          CZERO   000566R  020    D.BXNL= 000007          D.$PLU= 000000
ASCQUO= 000042           CHRNEG= 000100          C.ALOC= 000162          D.BXPP= 000007          D.$RES= 000000
ATAN1   001356R  002     CHRPDL= 000271          C.BLOK= 000064          D.BXPW= 000007          D.$SUB= 000000
ATAN2   001506R  002     CHRPLU= 000055          C.BLSZ= 000140          D.BXRL= 000007          D.1COM= 000001
ATAN3   001516R  002     CHRPWR= 000120          C.BUFF= 177760          D.CIRC= 000001          D.1EXP= 000002
A.ALEN= 000004           CHRQST= 000121          C.BUFL= 177762          D.COMB= 000001          D.1ROT= 000003
A.ARG  = 000040          CHRQUO= 000113          C.CCNT= 177766          D.COMP= 000001          EDICHR  005063R  025
A.ASB  = 000200          CHRRBR= 000047          C.CPTR= 177764          D.CONT= 000006          EDICNT  005064R  025
A.BITS= 000003           CHRRPR= 000042          C.CZ   = 001000         D.CTLM= 000000          EDIFLG  005062R  025
A.CON  = 000020          CHRSEM= 000074          C.DIG  = 000040         D.DEAL= 000003          EDIT   = 000003
A.DIMS= 000002           CHRSMO= 000112          C.FASW= 020000          D.DECD= 000002          EIS    = 000000
A.DIM1= 000006           CHRSPA= 000040          C.FLEN= 000010          D.DIV  = 000000         ELESIZ  005132R  025
A.NUM  = 000006          CHRUPA= 000131          C.FLSH= 004000          D.DMD  = 000010         EMPTY = ****** GX
A.PTR  = 000000          CHR.BS= 000010          C.FRC  = 000001         D.DQAD= 177777          ENDCOR  000004R  025
A.TMP  = 000010          CHR.CC= 000003          C.IDEN= 000010          D.DROP= 000002          EOF    = 104475
BADDIR= 104463           CHR.CR= 000015          C.NAME= 000102          D.ENCD= 000002          ERRMAX= 000166
BADERR= 104437           CHR.CZ= 000032          C.NOBF= 000400          D.EQ   = 000000         ERRNUM  004772R  025
BLARGC= ****** GX        CHR.GT= 000046          C.NUM  = 000020         D.EXPD= 000002          ERRPTR  004770R  025
BLZBUB  005116R  025     CHR.LF= 000012          C.OP   = 000004         D.FGET= 000005          ERR0    004774R  025
BRSELF  005046R  025     CHR.LT= 000043          C.OPN  = 010000         D.FMT  = 000004         EXEBUF  005054R  025
CCNT    005072R  025     CHR.SI= 000017          C.RDLN= 177776          D.FPNT= 000005          EXPBIG  000356R  002
CHANEX  003060R  025     CHR.SL= 000057          C.REWT= 002000          D.FPUT= 000005          EXP1    000266R  002
CHAN00  000256R  025     CHR.SO= 000016          C.RND  = 000010         D.GE   = 000000         E$CT    005056R  025
CHAN01  000330R  025     CHR.TB= 000011          C.SBIT= 177756          D.GT   = 000000         E$PT    005060R  025
CHAN02  000512R  025     CHR.UL= 000106          C.SPFX= 000100          D.IBM  = 000002         FA.APD= 000100
CHAN03  000674R  025     CHR..A= 000141          C.SPGL= 000200          D.IND  = 000002         FA.CRE= 000010
CHAN04  001056R  025     CHR..E= 000145          C.SPLX= 000001          D.INNR= 000013          FA.DLK= 001000
CHAN05  001240R  025     CHR..S= 000163          C.SPRS= 000002          D.INTR= 000006          FA.ENB= 100000
CHAN06  001422R  025     CHR..T= 000164          C.SRC  = 177774         D.LE   = 000000         FA.EXC= 002000
CHAN07  001604R  025     CHR..U= 000165          C.SRCP= 000024          D.LOG  = 000001         FA.EXT= 000004
CHAN08  001766R  025     CHR..Z= 000172          C.TT   = 000004         D.LT   = 000000         FA.NSP= 000100
CHAN09  002150R  025     CHR..0= 000060          C.WIDC= 177770          D.MAX  = 000000         FA.NSR= 000200
CHAN10  002332R  025     CHR..1= 000061          C.WIDR= 177772          D.MEM  = 000002         FA.POS= 010000
CHAN11  002514R  025     CH.AND= 000001          DEBUG  = 000000         D.MIN  = 000000         FA.RD  = 000001
CHAN12  002676R  025     CIRTAB  000000R  002    DEFERR= 104405          D.MUL  = 000000         FA.RWD= 004000
CHAN13  000022R  023     CLOSE0  002224RG 020    DEVIL   005114R  025    D.NE   = 000000         FA.SEQ= 040000
CHAN14  000204R  023     CMDBUF  003250R  025    DEVNTF= 104470          D.NN   = 000000         FA.SHR= 000040
CHNOPN= 104471           CMDMAX= 000010          DFXDIM= ****** GX       D.NR   = 000000         FA.TMP= 000020
CHRAND= 000051           CNEG4   000542R  020    DIGITS  000006R  025    D.OR   = 000000         FA.WCK= 020000
CHRBEL= 000007           CNTSIZ  005130R  025    DIMAX  = 000020         D.PLUS= 000000          FA.WRT= 000002
CHRBKR= 000133           COMBIN  005076R  025    DIMBLK  003112R  025    D.PSUB= 000006          FD.BLK= 000010
CHRBOL= 000200           COMCNT  005126R  025    DIMPTR  003114R  025    D.PWR  = 000001         FD.CCL= 000002
CHRBOX= 000114           COMERR= 104425          DIMTYP  003116R  025    D.QQAD= 000000          FD.CON= 020000
CHRCMA= 000054           COMPTR  005122R  025    DIRFUL= 104466          D.QUAD= 000000          FD.CR  = 000002
CHRCOL= 000076           COMP1   001016R  020    DOMERR= 104417          D.RES  = 000000         FD.DIR= 000010
CHRDEL= 000107           COMRTN  003156R  025    DPTERR= 104416          D.RHO  = 000000         FD.FTN= 000001
CHRDIV= 000053           COMSKP  005124R  025    DYDCNT  003224R  025    D.ROT  = 000003         FD.F11= 040000
CHRDLR= 000176           CORPTR  005120R  025    DYDTAB= ****** GX       D.SCON= 000006          FD.INS= 000010
CHRDLT= 000110           CPOS4   000574R  020    D$CIRC  000464RG 020    D.SDIF= 000006          FD.ISP= 002000
CHRDOT= 000056           CS.ASS= 000200          D$COMB  000614RG 020    D.SUB  = 000000         FD.MNT= 100000
```

SYMBOL TABLE

```
FD.OSP= 004000          F.BKDN= 000026          F.R2  = 000014          LARGSF  003172R 025 M.CHR = 000202
FD.PLC= 000004          F.BKDS= 000020          F.R3  = 000016          LARGTY  003170R 025 M.CMT = 000206
FD.PRN= 000004          F.BKEF= 000050          F.R4  = 000020          LARGVA  003200R 025 M.DOUT= 000006
FD.PSE= 010000          F.BKP1= 000051          F.SEQN= 000100          LBINC   005100R 025 M.DQAD= 000006
FD.RAH= 000001          F.BKST= 000024          F.SPDV= 000072          LBUFF   003316R 025 M.ENCD= 000004
FD.RAN= 000002          F.BKVB= 000064          F.SPUN= 000074          LBUFLN= 000621          M.EXE = 000000
FD.REC= 000001          F.BR  = 000020          F.STBK= 000036          LCOUNT  005106R 025 M.EXP = 000001
FD.RWM= 000001          F.CHR = 000075          F.STK = 000022          LENERR= 104412          M.FACT= 000001
FD.SDI= 000020          F.CNTG= 000034          F.STOP= 000004          LENTAB  000054R 025 M.FIL = 000177
FD.SQD= 000040          F.CODE= 000010          F.STP = 000001          LEXPTR  003100R 025 M.FLR = 177777
FD.TTY= 000004          F.DFNB= 000046          F.SUSP= 000002          LFTOP   005066R 025 M.FMT = 000004
FD.WBH= 000002          F.DSPT= 000044          F.TO.I= ****** GX       LINC    005102R 025 M.GDUP= 000003
FF.CHR= 000005          F.DVNM= 000134          F.TRC = 000002          LINE    005016R 025 M.GDWN= 000003
FF.NV = 000003          F.EFBK= 000010          F.UNIT= 000136          LININC  005026R 025 M.GETI= 000006
FF.POE= 000002          F.EFN = 000050          F.URBD= 000020          LISIZE= 000020          M.IBM = 000003
FF.RWD= 000001          F.END = 002000          F.VAL = 177776          LITOP   005072R 025 M.IDN = 000000
FF.RWF= 000006          F.EOBB= 000032          F.VBN = 000064          LNLNG = 104541          M.INSP= 000006
FF.SPC= 000004          F.ERR = 000052          F.VBSZ= 000060          LN1     000156R 002 M.IOTA= 000000
FHEADR  005012R 025     F.FACC= 000043          GAMDON  005044R 020     LOCLCT  005036R 025 M.LIT = 000205
FHRET   005040R 025     F.FFBY= 000014          GAMMA   004630R 020     LOWCOR  000002R 025 M.LN  = 000001
FILNTF= 104467          F.FLAG= 000002          GAMTAB  002050R 002     LSTERP  004766R 025 M.MMD = 000010
FIS   = 000000          F.FLB = 000100          GETREG= ****** GX       LWIDTH  000070R 025 M.NEG = 000000
FISCON= 046000          F.FNAM= 000110          GETSM2= ****** GX       L.BITS= 000003          M.NOT = 000000
FIXDIM= ****** GX       F.FNB = 000102          GNOME   003102R 025     L.CNT = 000002          M.NUB = 000006
FIXPTR  003074R 025     F.FNDF= 000010          HASHC   005136R 025     L.PRNT= 000001          M.NUM = 000204
FLAGS   004764R 025     F.FTYP= 000116          HASHTB  005140R 025     L.PTR = 000000          M.OUTR= 000012
FLARGC= ****** GX       F.FUN = 000010          HSHMSK= 177601          L.PTR1= 000004          M.PITM= 000000
FLCBUF  003250R 025     F.FVER= 000120          HSHSIZ= 000200          L.SUBS= 000002          M.PRDS= 000006
FLINE   005014R 025     F.HEAD= 000006          IASH  = ****** GX       MACHIN= 000106          M.QQAD= 000000
FLOEXP  003270R 025     F.HIBK= 000004          ICHAN   005050R 025     MAXCHN= 000014          M.QUAD= 000000
FLONEG  003272R 025     F.LBL = 000020          ILFNAM= 104464          MONBLK  003120R 025     M.RAND= 000012
FLSWTB  000222R 025     F.LINE= 000012          ILLCHN= 104540          MONCVT  003132R 025     M.RAVL= 000000
FLTLEN= 000010          F.LOC = 000004          ILLSW = 104565          MONLEN  003130R 025     M.RCIP= 000000
FNINSI  005044R 025     F.LSTC= 000004          IMUL  = ****** GX       MONPTR  003122R 025     M.RDCT= 000011
FORIGN  000032R 025     F.LUN = 000042          INDERR= 104410          MONSFT  003126R 025     M.REVS= 000003
FO.APD= 000106          F.MBCT= 000054          INNTMP  003250R 025     MONTYP  003124R 025     M.ROLL= 000000
FO.MFY= 000002          F.MBC1= 000055          INNTM2  005016R 025     MONVAL  003134R 025     M.SATA= 000001
FO.RD = 000001          F.MBFG= 000056          INPTR   000064R 025     MSEXP   000722RG 020    M.SCAN= 000011
FO.UPD= 000006          F.MON = 000002          INSNAM= 104462          MSFACT  000756RG 020    M.SHAP= 000000
FO.WRT= 000016          F.NIL = 000001          INTOVR  001660R 020     MSLN    000742RG 020    M.SIGN= 000000
FPMODE  000042R 025     F.NPBD= 000024          IOERR = 104477          M.ABS = 000000          M.SUBR= 000001
FPP   = 000000          F.NREC= 000030          I.CNT = 177776          M.BOL = 000200          M.SYM = 000203
FPPLEN= 000004          F.OPN = 000010          I.DIM = 177772          M.BXAV= 000007          M.SYN = 000201
FREBLK  007400R 025     F.OVBS= 000030          I.LEN = 000012          M.BXCR= 000007          M.TRAN= 000012
FRESCT= ****** GX       F.PREV= 000000          I.OERR= 000062          M.BXCT= 000007          M.$ABS= 000000
FUZZ    000142R 025     F.PRNT= 100000          I.OLD = 177770          M.BXEX= 000007          M.$CEI= 000000
FXPTR1  003076R 025     F.PROT= 010000          I.PTR = 177774          M.BXFX= 000007          M.$FLR= 000000
F$EVAL= 000100          F.RACC= 000016          I.STR = 177766          M.BXIO= 000007          M.$IDN= 000000
F$EVI = 000200          F.RATT= 000001          KILL  = ****** GX       M.BXLC= 000003          M.$NEG= 000000
F$EXEC= 000400          F.RCNM= 000034          LABERR= 104406          M.BXNC= 000007          M.$SIG= 000000
F$FUNC= 001000          F.RCTL= 000017          LABLPT  005042R 025     M.BXNL= 000007          M.1RDT= 000011
F.ACTL= 000076          F.READ= 004000          LARGBL  003162R 025     M.BXPP= 000007          M.1REV= 000003
F.ALOC= 000040          F.REDF= 000040          LARGCV  003176R 025     M.BXPW= 000007          M.1SCN= 000011
F.BBFS= 000062          F.RES = 000040          LARGFL  003160R 025     M.BXRL= 000007          NAMBUF  003240R 025
F.BDS = 000070          F.RET = 177772          LARGLE  003174R 025     M.BXTS= 000007          NBRTRU= ****** GX
F.BGBC= 000057          F.RSIZ= 000002          LARGLN  003166R 025     M.BXWA= 000007          NB.DEV= 000200
F.BITS= 177774          F.RTYP= 000000          LARGPT  003164R 025     M.CEI = 177777          NB.DIR= 000100
```

APLOO1 MACRO M1113 08-FEB-80 09:10 PAGE 94-3
SYMBOL TABLE

```
NB.NAM= 000004          OMODE   000402R     023 RTOP    005070R     025 SY.OPN= 000002          WSDATE  000136R     025
NB.SD1= 000400          OPTMP0  005066R     025 RT11  = 000000          SY.PMT= 001000          WSFULL= 104403
NB.SD2= 001000          OPTMP1  005070R     025 R.BITS= 000003          SY.SRT= 000400          WSNAME  000074R     025
NB.SNM= 000040          OPTMP2  005072R     025 R.FIX = 000001          SY.SVC= 000004          WSTIME  000132R     025
NB.STP= 000020          OPTMP3  005074R     025 R.LCT = 000002          SY.SVL= 000200          $ABSF = ****** GX
NB.SVR= 000010          OPTMP4  005076R     025 R.LN  = 000002          S.AB  = 177776          $ADDD = ****** GX
NB.TYP= 000002          OPTMP5  005100R     025 R.LNUM= 000010          S.ASB = 100000          $ADDF = ****** GX
NB.VER= 000001          OPTMP6  005102R     025 R.NEXT= 000006          S.BFHD= 000020          $CFCC = ****** GX
ND.DEV= 000200          OPTMP7  005104R     025 R.OFF = 000000          S.BITS= 000002          $CLRF = ****** GX
ND.EXT= 000002          OPTMP8  005106R     025 R.PREV= 000004          S.CHR1= 000003          $CMPD = ****** GX
ND.NAM= 000004          OPTMP9  005110R     025 R.PTR = 000000          S.FATT= 000016          $CMPF = ****** GX
ND.PPN= 000100          OPTM10  005112R     025 R.SEQ = 000003          S.FDB = 000140          $DIVD = ****** GX
ND.SEX= 000000          OPTM14  005026R     025 R.VAR = 000002          S.FLB = 040000          $DIVF = ****** GX
ND.SES= 000020          ORIGIN  000030R     025 R6    =%000006          S.FNAM= 000006          $FIS  = ****** G
ND.SNM= 000000          OUTPTR  000066R     025 R7    =%000007          S.FNB = 000036          $FPP  = ****** G
ND.SNS= 000040          OVROPR= 000001              SAME  = ****** GX   S.FNBW= 000017          $FPPLN= ****** G
ND.SPN= 001400          PATCH   007344R     025 SATAN   001304RG    020 S.FNTY= 000004          $LDCFD= ****** GX
NEGSGN  005063R     025 PI    = ****** GX       SATAN1  001324R     020 S.FTYP= 000002          $LDCIF= ****** GX
NINPT1  003302R     025 PRECSN  000052R     025 SI    =%000005          S.LEN = 000002          $LDD  = ****** GX
NINPT2  003300R     025 RANDAC  003064R     025 SIDAMN= 104433          S.LINK= 177774          $LDEXP= ****** GX
NINSTK  003304R     025 RANDY   003062R     025 SIFRAM  003314R     025 S.NFEN= 000020          $LDF  = ****** GX
NINTMP  003310R     025 RANERR= 104411          SIN1    000036R     002 S.TYPE= 140000          $LDFPS= ****** GX
NINTYP  003306R     025 RARGBL  003204R     025 SIN2    000046R     002 S.VAL = 000000          $MACHN= ****** G
NOTAVL= 104472          RARGCT= ****** GX       SIST4K  006350R     025 TKDIMS  005072R     025 $MODD = ****** GX
NOTOPN= 104473          RARGCV  003220R     025 SITOP   006346R     025 TKFRAM  005036R     025 $MODF = ****** GX
NPRT1   004762R     025 RARGFL  003202R     025 SITP    007144R     025 TKLEFT  005070R     025 $MULD = ****** GX
NULL    007364R     025 RARGLE  003216R     025 SIWARN= 000200          TMODE   000403R     023 $MULF = ****** GX
NUMCHN= 000020          RARGLN  003210R     025 SIZTAB  000044R     025 TMPTAG= 000166          $NEGF = ****** GX
NUMDFL  003230R     025 RARGPT  003206R     025 SI.BIT= 177774          TTYBUF  004140R     025 $RSTS = ****** G
NUMEFL  003233R     025 RARGSF  003214R     025 SI.LEN= 177772          TTYLEN= 000622          $RSX  = ****** G
NUMESN  003236R     025 RARGTY  003212R     025 SI.RET= 177772          TURKEY= 104436          $RT11 = ****** G
NUMFLG  003230R     025 RARGVA  003222R     025 SI.VAL= 177776          TYPE  = 000166          $STCDI= ****** GX
NUMLEN  000062R     025 RBASE   005074R     025 SKPSIZ  005134R     025 T.BOL = 020000          $STCFI= ****** GX
NUMNEG  003232R     025 RCOUNR  005112R     025 SPHOLD  000004R     024 T.CON = 004000          $STD  = ****** GX
NUMQFL  003234R     025 RCOUNT  005110R     025 SPINIT  000002R     024 T.DIM = 000100          $STEXP= ****** GX
NUMRIG  003231R     025 REDCNT  005070R     025 STKINL= 000774          T.FUN = 010000          $STF  = ****** GX
NUMWID  003226R     025 REDIGN  005066R     025 STOP  = 104420          T.LIST= 000400          $STFPS= ****** GX
NUMZFL  003235R     025 REDOFF  005072R     025 SUBCNT  003104R     025 T.LIT = 000020          $SUBD = ****** GX
NYIERR= 104432          REENTR  003060R     025 SUBDCT  003106R     025 T.MON = 000010          $SUBF = ****** GX
N.BLSZ= 000036          REGTYP= 000002              SUBPTR  003110R     025 T.NIL = 000004          $TSTD = ****** GX
N.DESC= 000020          RESBIT  003142R     025 SYMB1   005344R     025 T.NUM = 002000          $TSTF = ****** GX
N.DEV = 000032          RESBLK= ****** GX       SYMF    005340R     025 T.OPFN= 000001          .ACOS   000366RG    002
N.DEVN= 000034          RESCVT  003152R     025 SYMPT1  005346R     025 T.OTHR= 040000          .ACOSH  004226RG    020
N.DID = 000024          RESERR= 104440              SYMTAB  005352R     025 T.SOP = 000002          .ASIN   000430RG    002
N.DVNM= 000032          RESLEN  003150R     025 SYMTBF= 104404          T.SPOP= 000040          .ASINH  004272RG    020
N.EXT = 000014          RESPTR  003140R     025 SYMTOP  005350R     025 T.TMP = 000200          .ATAN   000622RG    002
N.FID = 000000          RESSFT  003146R     025 SYMTP   006272R     025 T.VAL = 100000          .ATANH  004140RG    020
N.FNAM= 000006          RESTYP  003144R     025 SYMURN= 000054          T.VAR = 001000          .BOOLE= 000002
N.FTYP= 000014          RESULT  003136R     025 SYNERR= 104407          UPPER = 177400          .CHARA= 000000
N.FVER= 000016          RESVAL  003154R     025 SYSERR= 104402          UTNAME  000344R     023 .COS    002254RG    020
N.NAME= 000006          RINC    005104R     025 SYSTAT  000000R     024 VALERR= 104413          .COSH   001610RG    002
N.NEXT= 000022          RLTBAS  005066R     025 SY.BAD= 000001          VARBLE  007400R     025 .DEF. = 000002
N.PPN = 000024          ROTINC  005070R     025 SY.BRK= 100000          VARBLS  000000R     025 .EXP    003450RG    020
N.STAT= 000020          ROUND   000010R     025 SY.BS = 000010          VERNUM  000000R     025 .FACT   004320RG    020
N.UNIT= 000034          ROVER   003312R     025 SY.CLR= 000040          VRSION= 000002          .FLOAT= 000006
OBUFLN= 001000          RSTS  = 000000              SY.ERP= 000020          WIDTH   000072R     025 .FSRCB= ****** G
OCHAN   005052R     025 RSX   = 000115          SY.NOU= 000100          WROUND  000020R     025 .INTEG= 000004
```

```
APLO01  MACRO M1113  08-FEB-80 09:10  PAGE 94-4
SYMBOL TABLE

.LN     003012RG    020 .SIN    002270RG    020 .SQRT = ****** GX       .TANH   001672RG    002 ...TPC= 000020
.MOLUN  000404R     023 .SINH   001526RG    002 .TAN    002734RG    020 ...GBL= 000000

. ABS.  000000      000
        000000      001
TABLES  002230      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    005102      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  14008 WORDS  ( 55 PAGES)
DYNAMIC MEMORY:  15534 WORDS  ( 59 PAGES)
ELAPSED TIME:  00:06:06
APLO01,DM1:[305,301]APLO01/-SP=APLRX1,APLDEF,APLO01,APLOP,END
```

FILEID**APL002

```
  AAAAAA      PPPPPPPP      LL              000000        000000        222222
  AAAAAA      PPPPPPPP      LL              000000        000000        222222
AA      AA    PP      PP    LL            00    00      00    00    22      22
AA      AA    PP      PP    LL            00    00      00    00    22      22
AA      AA    PP      PP    LL            00    00      00  0000          22
AA      AA    PP      PP    LL            00    00      00  0000          22
AA      AA    PPPPPPPP      LL            00    00      00  00  00      22
AA      AA    PPPPPPPP      LL            00    00      00  00  00      22
AAAAAAAAAA    PP            LL            00    00      0000    00    22
AAAAAAAAAA    PP            LL            00    00      0000    00    22
AA      AA    PP            LL            00    00      00      00      22            ....
AA      AA    PP            LL            00    00      00      00      22            ....
AA      AA    PP            LLLLLLLLLL      000000        000000        2222222222    ....
AA      AA    PP            LLLLLLLLLL      000000        000000        2222222222    ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 09:16
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    53- 1639    MIXED FUNCTION           INDEX OF
```

APL002 MACRO M1113 08-FEB-80 09:16
TABLE OF CONTENTS

55- 1792	MIXED FUNCTION	FIRST CO-ORDINATE EXPANSION
55- 1806	MIXED FUNCTION	EXPANSION
56- 1867	MIXED FUNCTION	DROP
56- 1912	MIXED FUNCTION	TAKE
61- 2657	MIXED FUNCTION	ENCODE
62- 2859	MIXED FUNCTION	DECODE
67- 3867	MIXED FUNCTION	MEMBERSHIP
74- 4962	MIXED FUNCTION	I BEAM

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 1


      1        000000                  RT11    =  0          ;NOT RT-11
      2        000000                  RSTS    =  0          ;NOT RSTS/E
      3        000115                  RSX     = 'M          ;RSX-11/M
      4
      5        000000                  EIS     =  0          ;NO EIS
      6        000000                  FIS     =  0          ;NO FIS
      7        000000                  FPP     =  0          ;NO FPP
      8
      9        000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOO2  MACRO M1113  08-FEB-80 09:16  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  APLOO2
                                .IDENT  \V02.03\

                                ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
                                ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
                                ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
                                ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
                                ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO2 IS:
                                ;
                                ;       APLOO2 < ...,APLOO2,APLOP
                                ;

000002                          OVROPR= 2                       ;OVERLAY SPECIFIED OPERATORS
```

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                                ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                                ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                                ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                                ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                                ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                                ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                                ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                                ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                                ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                                ;       INTACT!
                                ;--



                                ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                                ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                                ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                                ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                                ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                                ;MACROS.

                                .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-D.'NAME
                                        .PSECT  CODE
                               D$'NAME::
                                .ENDM

                                .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-M.'NAME
                                        .PSECT  CODE
                                        .IF NE NUM
                               M$'NAME::
                                    .ENDC
                                .ENDM

                                .MACRO  LIST
                                        .ENDC
                                .ENDM

                                .MACRO  PRMRES
                                        .IF EQ 1
                                .ENDM

                                                ;************ ASSEMBLING THE FILE!!!!!
 160 007400                    PRMRES
 199 007400                            LIST
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 47
DEFINITIONS     ERRORS

    201 007400                          DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,,FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 47-1
DEFINITIONS     ERRORS

    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

```
641 007400                          DOPINS  AND,42,T.SOP,,BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,,BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,,BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,,BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,,CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,,CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.SOP,,CHARACTER+CS.IDN
780 007400                                  LIST
781
```

```
783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
786 007400                                  LIST
787
788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
791 007400                                  LIST
792
793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
799 007400                                  LIST
800
801
802 007400                          MOPINS  $IDN,-2
808 007400                                  LIST
809
810
811 007400                          MOPINS  $FLR,-14
855 007400                                  LIST
856
857
858 007400                          MOPINS  $CEI,-16
868 007400                                  LIST
869
870
871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
878 007400                                  LIST
879
880 007400                          MOPINS  $NEG,-4
887 007400                                  LIST
888
889
890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
901 007400                                  LIST
902
903 007400                          MOPINS  $SIG,-6
914 007400                                  LIST
915
916
917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
928 007400                                  LIST
929
930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
939 007400                                  LIST
940
941
942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
949 007400                                  LIST
950
951
952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
959 007400                                  LIST
960
961 007400                          PRMRES
966 007400                                  LIST
967
968
969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
977 007400                                  LIST
978
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

F 7

DEFINITIONS ERRORS

```
1059 007400                          DOPINS  DEAL,100
1187 007400                                  LIST
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 51
DEFINITIONS     ERRORS

   1189 007400                          MOPINS  SHAP,102
   1211 007400                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

```
                                        MOPINS  IOTA,72
                                        LIST
                                        DOPINS  IND,72
                                        .SBTTL  MIXED FUNCTION          INDEX OF

000000                                  CALL    GETREG          ;GET SOME REGISTERS AND BITS
000004  122763  000001  000002          CMPB    #1,A.DIMS(R3)   ;LEFT A VECTOR?
000012  002001                          BGE     1$              ;YES, FINE
000014  104411                          RANERR                  ;BAD LEFT ARG
000016  042700  177771          1$:     BIC     #^C<A.NUM>,R0   ;CLEAR OUT GARBAGE
000022  010002                          MOV     R0,R2           ;SOURCE TYPE IS BIGGER OF TWO
000024  001004                          BNE     2$              ;IT IS NUMERIC
000026  042701  177771                  BIC     #^C<A.NUM>,R1   ;CLEAR OUT GARBAGE
000032  001066                          BNE     8$              ;TYPES DON'T MATCH
000034  000406                          BR      3$              ;...GO

000036  042701  177771          2$:     BIC     #^C<A.NUM>,R1   ;CLEAR GARBAGE BITS
000042  001462                          BEQ     8$              ;NOT SAME, ERROR
000044  020201                          CMP     R2,R1           ;WHICH TYPE BIGGER?
000046  002001                          BGE     3$              ;EVIDENTLY, THE 1ST
000050  010102                          MOV     R1,R2           ;NO, IT'S THE 2ND
000052  016237  000000G 005122' 3$:     MOV     COMCT(R2),COMPTR;COMPARE ROUTINE
000060  006302                          ASL     R2              ;TIMES 2
000062  006302                          ASL     R2              ;  "   4
000064  060200                          ADD     R2,R0           ;LEFT ARG FETCH ROUTINE
000066  016037  000000G 003176'         MOV     LARGCT(R0),LARGCV ;SO HERE IT IS
000074  060201                          ADD     R2,R1           ;RIGHT ARG FETCH ROUTINE
000076  016137  000000G 003220'         MOV     RARGCT(R1),RARGCV ;SO, HERE IT IS
000104  010400                          MOV     R4,R0           ;COPY POINTER TO RIGHT ARG
000106  012701  000004                  MOV     #.INTEGER,R1    ;RESULT TYPE IS INTEGER
000112                                  CALL    GETSM2          ;GET SAME SIZE ARRAY
000116                                  PUSH    A.ALEN(R3)      ;SAVE LENGTH OF LEFT ARG
000122  011001                          MOV     @R0,R1          ;POINT TO 1ST ELEMENT
000124  011403                          MOV     @R4,R3          ;POINT TO 1ST ELEMENT OF RIGHT
000126  016404  000004                  MOV     A.ALEN(R4),R4   ;GET # OF ELEMENTS
000132  001424                          BEQ     7$              ;NOTHING THERE
                                ;
                                ;PROCESS NEXT ARG IN RIGHT ARRAY
                                ;
000134                          4$:     CALL    @RARGCV         ;FETCH RIGHT ARG
000140  013702  003164'                 MOV     LARGPT,R2       ;POINT TO 1ST ELEMENT IN LEFT ARG
000144  011600                          MOV     (SP),R0         ;GET COUNT FOR INNNER LOOP
000146  001407                          BEQ     6$              ;NOTHING THERE, EVIDENTLY
                                ;
                                ; INNER LOOP - DO COMPARES
                                ;
000150                          5$:     CALL    @LARGCV         ;GET LEFT ARG
000154                                  CALL    @COMPTR         ;COMPARE WITH RIGHT ARG
000160  001402                          BEQ     6$              ;IF EQUAL, STOP LOOP
000162                                  SOB     R0,5$           ;LOOP FOR EACH ELEMENT
                                ;
                                ;MARK A HIT IN THE RESULT ARRAY
```

```
1688                                    ;
1689 000166  161600                6$:  SUB     (SP),R0         ;THE NEGATIVE OF POSITION
1690 000170  005400                     NEG     R0              ;NOW, THE POSITION IN THE ARRAY
1691 000172  063700  000030'            ADD     ORIGIN,R0       ;PLUS THE ORIGIN
1692 000176  010021                     MOV     R0,(R1)+        ;STORE THE RESULT
1693 000200                             SOB     R4,4$           ;LOOP FOR ALL
1694 000204  005726                7$:  TST     (SP)+           ;POP ELEMENT
1695 000206                             RETURN
1696
1697 000210  104417                8$:  DOMERR                  ;UNEQUAL TYPES
1699 000212                             LIST
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 54
MIXED FUNCTION          INDEX OF

  1701 000212                          DOPINS  1COM,124
  1709 000212                                  LIST
  1710
  1711
  1712
  1713
  1714
  1715 000212                          DOPINS  COMP,60,T.DIM
  1789 000212                                  LIST
```

```
1791 000212                                DOPINS  1EXP,130
1792                                       .SBTTL  MIXED FUNCTION          FIRST CO-ORDINATE EXPANSION
1793
1794 000212                                        CALL    GETREG          ;LOAD SOME REGISTERS
1795 000216  012702  000001                       MOV     #1,R2           ;USE DIM 1
1796 000222                                        CALL    FIXDIM          ;SET UP DIM POINTER
1797 000226  000137  000236'                       JMP     EXPND1          ;DO EXPANSION
1799 000232                                        LIST
1800
1801
1802
1803
1804
1805 000232                                DOPINS  EXPD,62,T.DIM
1806                                       .SBTTL  MIXED FUNCTION          EXPANSION
1807
1808 000232                                        CALL    DFXDIM          ;SET UP DIMENSION POINTER
1809 000236  013701  003170'               EXPND1: MOV     LARGTY,R1       ;GET LEFT ARG TYPE
1810 000242  001001                               BNE     1$              ;YES
1811 000244  104417                               DOMERR                  ;NO, ERROR
1812 000246  016137  000000G 003176'       1$:     MOV     BLARGC(R1),LARGCV ;LEFT ARG FETCH
1813 000254  126327  000002  000001               CMPB    A.DIMS(R3),#1   ;IS LEFT SCALAR OR VECTOR?
1814 000262  003401                               BLE     2$              ;YES
1815 000264  104411                               RANERR                  ;NO, ERROR
1816 000266                                2$:     CALL    NBRTRU          ;GET THE # OF 1'S IN LEFT
1817 000272  020177  005122'                       CMP     R1,@COMPTR      ;IS IT THE SAME AS THE DIM SIZE ON RIGHT?
1818 000276  001404                               BEQ     3$              ;YES! GOOD
1819 000300  105764  000002                       TSTB    A.DIMS(R4)      ;IS THE RIGHT SIDE A SCALAR?
1820 000304  001406                               BEQ     12$             ;YES...
1821 000306  104412                               LENERR                  ;LENGTH ERROR OF TOO MANY OR TOO FEW 1'S
1822 000310  020163  000004               3$:     CMP     R1,A.ALEN(R3)   ;ALL 1'S?
1823 000314  001002                               BNE     12$             ;NO, DO SOME EXPANSIONS
1824 000316  000137  000000G                       JMP     SAME            ;LONG JUMP TO 'SAME'
1825 000322  016301  000004               12$:    MOV     A.ALEN(R3),R1   ;GET NEW DIM LENGTH
1826 000326  116446  000002                       MOVB    A.DIMS(R4),-(SP);SAVE DIM COUNT
1827 000332  112746  000040                       MOVB    #CHRSPA,-(SP)   ;STORE A SPACE FOR NULL
1828 000336  005737  003212'                       TST     RARGTY          ;IS IT NUMERIC?
1829 000342  001401                               BEQ     10$             ;NO
1830 000344  005016                               CLR     (SP)            ;YES, SAY NUMERIC
1831 000346                                10$:    CALL    RESBLK          ;CREATE THE RESULT BLOCK
1832 000352  005764  000004                       TST     A.ALEN(R4)      ;Anything there?
1833 000356  001447                               BEQ     11$             ;No, finished
1834                                       ;
1835                                       ;DO THE ACTUAL EXPAND
1836                                       ;
1837 000360  105766  000002                       TSTB    2(SP)           ;RIGHT ARG A SCALAR?
1838 000364  001007                               BNE     30$             ;NO...
1839 000366  013766  005134'  000002              MOV     SKPSIZ,2(SP)    ;YES, SO SET IT
1840 000374  112764  000001  000002              MOVB    #1,A.DIMS(R4)   ;RESULT IS A VECTOR
1841 000402  000402                               BR      31$             ;AND SKIP AROUND
1842 000404  005066  000002               30$:    CLR     2(SP)           ;NO FIXUP WANTED
1843 000410  013704  003206'               31$:    MOV     RARGPT,R4       ;POINT TO THE RIGHT ARG'S DATA
1844 000414  013702  003164'               4$:     MOV     LARGPT,R2       ;POINT TO THE LEFT ARG'S DATA
1845 000420  013703  003166'                       MOV     LARGLN,R3       ;GET THE LEN OF THE LEFT ARG
1846 000424  013701  005134'               5$:     MOV     SKPSIZ,R1       ;GET 'SIZE OF GROUP'
1847 000430                                        CALL    @LARGCV         ;CHECK LEFT ELEMENT
1848 000434  105737  003200'                       TSTB    LARGVA          ;IS IT ZERO?
```

```
1849 000440  001406                    BEQ     8$              ;YES
1850 000442  112420            6$:     MOVB    (R4)+,(R0)+     ;COPY DATA
1851 000444                            SOB     R1,6$           ;ALL OF IN THE GROUP, PLEASE!
1852 000450  166604  000002            SUB     2(SP),R4        ;CHEAT FOR SCALAR
1853 000454  000403                    BR      9$              ;SEE IF MORE IN LEFT
1854
1855 000456  111620            8$:     MOVB    (SP),(R0)+      ;EXPAND WITH FILLER
1856 000460                            SOB     R1,8$           ;FILL IN WHOLE GROUP!
1857
1858 000464                    9$:     SOB     R3,5$           ;MORE IN LEFT?
1859 000470  005337  005126'           DEC     COMCNT          ;DONE WITH ALL GROUPS
1860 000474  001347                    BNE     4$              ;NO, SO DO SOME MORE
1861 000476  022626            11$:    CMP     (SP)+,(SP)+     ;CLEAN UP STACK
1862 000500                            RETURN                  ;YES, FINALLY!! (FINIS)
1864 000502                            LIST
```

```
1866 000502                                DOPINS  DROP,104
1867                                       .SBTTL  MIXED FUNCTION          DROP
1868
1869 000502                                        CALL    TAKCHK          ;CHECK ON ARGS
1870 000506  010304                                MOV     R3,R4           ;COPY LEFT ARRAY
1871 000510                                        CALL    SAME            ;WELL, DO IT!
1872 000514  014504                                MOV     -(SI),R4        ;SAVE POINTER
1873 000516  014503                                MOV     -(SI),R3        ;SAVE BITS
1874 000520  014502                                MOV     -(SI),R2        ;SAVE RETURN
1875 000522  020465  177776                        CMP     R4,SI.VAL(SI)   ;SAME AS OLD?
1876 000526  001002                                BNE     10$             ;NO, ALL'S OK
1877 000530  005065  177774                        CLR     SI.BIT(SI)      ;DON'T REALLY DELETE IT..
1878 000534                                10$:    CALL    KILL            ;DO AWAY WITH ORIGINAL LEFT
1879 000540  010225                                MOV     R2,(SI)+        ;STACK RETURN
1880 000542  010325                                MOV     R3,(SI)+        ;STACK BITS
1881 000544  010425                                MOV     R4,(SI)+        ;STACK POINTER
1882
1883 000546  010437  003162'                       MOV     R4,LARGBL       ;NEW LEFT ARG POINTER
1884 000552  011404                                MOV     (R4),R4         ;POINT TO FIRST ELEMENT
1885 000554  010437  003164'                       MOV     R4,LARGPT       ;SAVE FIRST ELEMENT POINTER
1886 000560  013703  003204'                       MOV     RARGBL,R3       ;GET POINTER TO RIGHT ARG BLOCK
1887 000564  062703  000006                        ADD     #A.DIM1,R3      ;BUMP POINTER TO DIMS
1888 000570  013702  003166'                       MOV     LARGLN,R2       ;GET LENGTH OF LEFT
1889                                       ;
1890                                       ;CONVERT 'DROP' TO 'TAKE'
1891                                       ;
1892 000574  012300                        1$:     MOV     (R3)+,R0        ;GET RIGHT
1893 000576  005714                                TST     (R4)            ;GET LEFT ARG VALUE
1894 000600  002402                                BLT     2$              ;NEGATIVE LEFT ARG
1895 000602  161400                                SUB     (R4),R0         ;POSITIVE, SUBTRACT
1896 000604  000401                                BR      3$              ;GO...
1897 000606  061400                        2$:     ADD     (R4),R0         ;NEGATIVE, ADD
1898 000610  002001                        3$:     BGE     4$              ;NO
1899 000612  005000                                CLR     R0              ;CLEAR DIFF
1900 000614  005714                        4$:     TST     (R4)            ;CHECK SIGN OF LEFT
1901 000616  003401                                BLE     5$              ;YES!
1902 000620  005400                                NEG     R0              ;NEGATE DIFF
1903 000622  010024                        5$:     MOV     R0,(R4)+        ;SAVE AS NEW ARG FOR 'TAKE'
1904 000624                                        SOB     R2,1$           ;BE SURE TO DO WHOLE LEFT ARG
1905 000630  000137  000640'                       JMP     TAKEIT          ;DO THE TAKING
1907 000634                                        LIST
1908
1909
1910
1911 000634                                DOPINS  TAKE,114
1912                                       .SBTTL  MIXED FUNCTION          TAKE
1913
1914 000634                                        CALL    TAKCHK          ;CHECK ON VALIDITY OF ARGS
1915                                       ;        BR      TAKEIT          ;FALL THROUGH TO 'TAKEIT'
1916
1917                                       ;++
1918                                       ;'TAKEIT' CREATES PARAMETER FRAMES FOR, AND INITIATES 'TAKER',
1919                                       ;WHICH DOES THE ACTUAL DATA MOVING.  THE FRAMES CREATED BY 'TAKEIT'
1920                                       ;ARE USED TO DECIDE WHAT AND WHERE TO COPY, AND WHEN TO SKIP FOR
1921                                       ;EACH DIM.  ONE FRAME IS CREATED FOR EACH DIM OF THE RESULT, PLUS
1922                                       ;ONE EXTRA FRAME FOR SIGNALLING THE END ON THE RECURSION CHAIN
1923                                       ;AND THE LAST DIM HAS BEEN REACHED.  THE FRAMES ARE:
```

```
1924
1925                                    ;
1926                                    ;       ******************
1927                                    ;       *  RESULT SKIP   *      # OF BYTES TO SKIP IN THE RESULT
1928                                    ;       ******************
1929                                    ;       *  SOURCE SKIP   *      # OF BYTES TO SKIP IN THE SOURCE
1930                                    ;       ******************
1931                                    ;       *    GROUPS      *      # OF TIMES TO RECURSE ON LOWER FRAME
1932                                    ;       ******************
1933                                    ;
1934                                    ;IF THE SKIPS ARE NEGATIVE, THEY ARE TO BE DONE BEFORE THE COPYING.
1935                                    ;IF THEY ARE POSITIVE, DO THEM AFTERWORDS.  ON THE LAST FRAME (END-OF-
1936                                    ;LIST FRAME) THE NUMBER OF TIMES IT IS TO BE RECURSED, IS THE NUMBER
1937                                    ;OF ELEMENTS THAT ARE TO BE COPIED.  (IE. THE DATA ELEMENTS ARE THE
1938                                    ;TREE LEAVES AND 'TAKEIT' IS THE GUY WHO PLANTS THE ROOTS)
1939                                    ;--
1940
1941 000640                             TAKEIT: CALL    GETREG          ;GET SOME REGISTERS LOADED
1942 000644  016301  000004                     MOV     A.ALEN(R3),R1   ;LEN OF LEFT (# OF DIMS IN RIGHT)
1943 000650                                     PUSH    R1              ;SAVE # OF DIMS
1944 000652  005201                             INC     R1              ;INC TO ADD 1 EXTRA FRAME
1945 000654                                     MUL     #6,R1           ;MAKE LEN OF FRAME AREA
1946 000670  010100                             MOV     R1,R0           ;COPY LEN
1947 000672                                     CALL    GETCOR          ;GET CORE FOR FRAMES
1948 000676  010037  005066'                    MOV     R0,TKFRAM       ;SAVE POINTER TO FRAMES BLOCK
1949 000702  060100                             ADD     R1,R0           ;POINT TO END OF FRAMES BLOCK
1950 000704  016302  000004                     MOV     A.ALEN(R3),R2   ;GET LEN OF LEFT (# OF DIMS OF RIGHT)
1951 000710  005302                             DEC     R2              ;ONE LESS FOR CORRECTNESS OF POINTERS
1952 000712  011303                             MOV     (R3),R3         ;POINT TO RIGHT DATA
1953 000714  062704  000006                     ADD     #A.DIM1,R4      ;POINT TO LEFT DIMS
1954 000720  006302                             ASL     R2              ;MULTIPLY BY 2 (WORDS)
1955 000722  060204                             ADD     R2,R4           ;POINT TO LAST DIM
1956 000724  060203                             ADD     R2,R3           ;POINT TO LAST LEFT ARG
1957 000726  010337  005070'                    MOV     R3,TKLEFT       ;SAVE POINTER TO LEFT ARGS
1958 000732  010437  005072'                    MOV     R4,TKDIMS       ;SAVE POINTER TO RIGHT DIMS
1959 000736  012701  000001                     MOV     #1,R1           ;INITIALIZE SOURCE SKIP PRODUCT TO 1
1960 000742  010103                             MOV     R1,R3           ;INITIALIZE RESULT SKIP TO 1 TOO
1961 000744  005040                             CLR     -(R0)           ;ZERO LAST FRAME
1962 000746  005040                             CLR     -(R0)           ;SKIPS TOO!
1963 000750  005040                             CLR     -(R0)           ;BOTH OF 'EM
1964
1965                                    ;CREATE FRAMES
1966                                    ;
1967 000752  017704  005070'            1$:     MOV     @TKLEFT,R4      ;GET # FROM LEFT
1968 000756  005040                             CLR     -(R0)           ;ZERO # TO TAKE
1969 000760  005040                             CLR     -(R0)           ;ASSUME NO SOURCE SKIPS
1970 000762  005040                             CLR     -(R0)           ;AND NO RESULT SKIPS
1971 000764  010402                             MOV     R4,R2           ;COPY ARG
1972 000766  100001                             BPL     3$              ;NEGATIVE?
1973 000770  005402                             NEG     R2              ;YES, MAKE POSITIVE
1974 000772                             3$:     PUSH    <R1,R3>         ;SAVE CURRENT SKIP PRODUCTS
1975 000776  167702  005072'                    SUB     @TKDIMS,R2      ;SUBTRACT OFF RIGHT DIM
1976 001002  013703  003212'                    MOV     RARGTY,R3       ;GET RIGHT ARG TYPE
1977 001006                                     ASH     SIZTAB(R3),R2   ;MULTIPLY BY PRECISION
1978 001022  011603                             MOV     @SP,R3          ;RESTORE R3
1979 001024  005702                             TST     R2              ;IS THIS A NEGATIVE SKIP?
1980 001026  100011                             BPL     2$              ;NO
```

```
                                        ;
                                        ;LOAD SOURCE SKIP
                                        ;
001030  005402                         NEG     R2              ;MAKE POSITIVE
001032                                  MUL     R2,R1           ;MULTIPLY BY NUMBER OF ELEMENTS TO SKIP
001044  010160  000002                  MOV     R1,2(R0)        ;SAVE AS SKIP FOR FRAME
001050  000406                         BR      10$
                                        ;
                                        ;LOAD RESULT SKIP
                                        ;
001052                          2$:     MUL     R2,R3           ;ULTIPLY BY NUMBER OF ELEMENTS TO SKIP
001064  010310                         MOV     R3,(R0)         ;SAVE AS # OF BYTES TO SKIP IN RESULT
                                
001066  005704                 10$:    TST     R4              ;SKIP BEFORE COPY?
001070  100004                         BPL     4$              ;NO, AFTER
001072  005404                         NEG     R4              ;MAKE LEFT ARG POSITIVE
001074  005410                         NEG     (R0)            ;NEGATE SKIPS
001076  005460  000002                 NEG     2(R0)           ;ALL TWO OF EM
001102                          4$:     POP     <R3,R1>         ;RESTORE SKIP PRODUCTS
001106                                  MUL     R4,R3           ;MULTIPLY BY RESULT SKIP PRODUCT
001120  013702  005072'                 MOV     TKDIMS,R2       ;GRAB CURRENT DIM POINTER
001124                                  MUL     (R2),R1         ;MULTIPLY FOR NEW SOURCE SKIP PRODUCT
001136  021204                         CMP     (R2),R4         ;IS DIM LESS THEN LEFT ARG?
001140  103001                         BHIS    6$              ;NO
001142  011204                         MOV     (R2),R4         ;USE DIM FOR NUMBER TO RECURSE
001144  162737  000002  005072' 6$:    SUB     #2,TKDIMS       ;SAVE NEW DIM POINTER
001152  010460  000004                 MOV     R4,4(R0)        ;LOAD # TO RECURSE (# OF SUB-GROUPS)
001156  162737  000002  005070'        SUB     #2,TKLEFT       ;MOVE TO NEXT LEFT #
001164  005316                         DEC     (SP)            ;DONE WITH LEFT ARG
001166  003271                         BGT     1$              ;NO
                                        ;
                                        ;CREATE RESULT BLOCK
                                        ;
001170  013704  003204'         XX3:    MOV     RARGBL,R4       ;POINT TO LEFT ARG DATA
001174  010301                         MOV     R3,R1           ;COPY RESULT LEN

001176  013700  003212'                 MOV     RARGTY,R0       ;GET RIGHT ARG TYPE
001202                                  ASH     SIZTAB(R0),R3   ;MULTIPLY BE PRECISION
001216  116400  000002                 MOVB    A.DIMS(R4),R0   ;GET # OF DIMS
001222                                  CONINC  R0              ;CHEAT FOR SCALAR
001226  006300                         ASL     R0              ;    TIMES 2
001230  062700  000006                 ADD     #A.DIM1,R0      ;OVERHEAD FOR OTHER STUFF
001234                                  PUSH    R0              ;SAVE SIZE FOR LATER
001236  060300                         ADD     R3,R0           ;PREP FOR GET
001240                                  CALL    GETFRA          ;Get result, SI frame for it
001244  062620                         ADD     (SP)+,(R0)+     ;ADD OFFSET TO DATA
001246  005724                         TST     (R4)+           ;POINT TO BITS
001250  032714  003000                 BIT     #A.NUM*400,(R4) ;NUMERIC?
001254  001003                         BNE     8$              ;Yes
001256  042765  002000  177774         BIC     #T.NUM,SI.BIT(SI);Say so on the SI
001264  112420                  8$:    MOVB    (R4)+,(R0)+     ;# OF DIMS
001266  001002                         BNE     11$             ;NOT A SCALAR
001270  105260  177777                 INCB    -1(R0)          ;NO SCALARS ALLOWED
001274  112710  000200          11$:    MOVB    #A.ASB,(R0)     ;ASSUME LITERAL
001300  153720  003212'                 BISB    RARGTY,(R0)+    ;SET RESULT TYPE BITS
001304  010120                         MOV     R1,(R0)+        ;LOAD LENGTH OF ARRAY
001306  013704  003162'                 MOV     LARGBL,R4       ;POINT TO LEFT BLOCK
```

```
2038 001312  016402  000006                MOV     A.DIM1(R4),R2   ;GET LEN OF LEFT
2039 001316  011404                        MOV     (R4),R4         ;POINT TO DATA
2040
2041 001320                        9$:     PUSH    (R4)+           ;GET # FROM LEFT
2042 001322  002001                        BGE     10$             ;WANT ABS VALUE
2043 001324  005416                        NEG     @SP             ;SO, NOW IT'S POSITIVE
2044 001326                        10$:    POP     (R0)+           ;FIX IT AS A DIM
2045 001330                                SOB     R2,9$           ;SEE IF MORE
2046
2047 001334  016500  177776                MOV     SI.VAL(SI),R0   ;Get pointer to result block
2048 001340  005760  000004                TST     A.ALEN(R0)      ;Is the result a null array?
2049 001344  001002                        BNE     12$             ;No
2050 001346  005726                        TST     (SP)+           ;Yes, Clean up the stack
2051 001350  000432                        BR      XX2             ;And get out of here fast
2052 001352  011000                12$:    MOV     (R0),R0         ;Now point to the result data
2053 001354  010002                        MOV     R0,R2           ;COPY POINTER
2054 001356  012703  000040                MOV     #CHRSPA,R3      ;ASSUME FILLING WITH SAPCES FOR ZEROING
2055 001362  005737  003212'               TST     RARGTY          ;NUMERIC?
2056 001366  001411                        BEQ     TAKE01          ;NO
2057 001370  013703  003212'               MOV     RARGTY,R3       ;GET RESULT TYPE
2058 001374                                ASH     SIZTAB(R3),R1   ;MULTIPLY BY PRECISION
2059 001410  005003                        CLR     R3              ;FILL WITH ZEROS
2060
2061 001412  110322                TAKE01: MOVB    R3,(R2)+        ;'ZERO' ARRAY
2062 001414                                SOB     R1,TAKE01       ;WHOLE THING
2063
2064 001420  005216                        INC     (SP)            ;REPEAT ONCE (CREATE ROOT SP FRAME)
2065 001422                                PUSH    <TKFRAM>        ;POINTER TO FIRST FRAME
2066 001426  013704  003206'               MOV     RARGPT,R4       ;POINT TO RIGHT DATA
2067 001432                                CALL    TAKER           ;CALL DATA MOVER
2068 001436  013700  005066'        XX2:    MOV     TKFRAM,R0       ;POINT TO PARAMETER FRAMES
2069 001442                                CALL    RETCOR          ;GET RID OF IT
2070 001446                                RETURN                  ;AND WE'RE ALL DONE!!!!!!
2071
2072                                ;+
2073                                ;'TAKCHK' CHECKS THE VALIDITY OF THE ARGS FOR 'TAKE' AND 'DROP'
2074                                ;-
2075
2076 001450                        TAKCHK: CALL    GETREG          ;GER SOME REGISTER VALUES
2077 001454  116302  000002                MOVB    A.DIMS(R3),R2   ;VECTOR OR SCALAR?
2078 001460  005302                        DEC     R2              ;;SCALAR OR VECTOR?
2079 001462  003023                        BGT     3$              ;NO, RANK ERROR
2080 001464  116402  000002                MOVB    A.DIMS(R4),R2   ;RIGHT SIDE # OF DIMS
2081 001470                                CONINC  R2              ;ADJUST FOR SCALAR
2082 001474  020263  000004                CMP     R2,A.ALEN(R3)   ;LEFT SAME LEN AS DIMS ON RIGHT?
2083 001500  001015                        BNE     4$              ;NO!
2084 001502  010502                        MOV     SI,R2           ;POINT TO LEFT ARG
2085 001504  012701  000004                MOV     #.INTEGER,R1    ;WE WANT TO BE INTEGER
2086 001510                                CALL    CONVERT         ;CONVERT TO INTEGER
2087 001514  016503  177776                MOV     SI.VAL(SI),R3   ;POINT TO ARG
2088 001520  010337  003162'               MOV     R3,LARGBL       ;SAVE POINTER
2089 001524  011337  003164'               MOV     @R3,LARGPT      ;AND POINTER TO DATA
2090 001530                                RETURN                  ;ALL OK! SO RETURN
2091
2092 001532  104411                3$:     RANERR                  ;RANK ERROR
2093 001534  104412                4$:     LENERR                  ;LENGTH ERROR
2094
```

```
2095
2096                                    ;TKFRAM= OPTMP0                 ;POINTER TO FRAME BLOCK FOR 'TAKEIT'
2097                                    ;TKLEFT= OPTMP1                 ;POINTER TO NEXT LEFT ELEMENT FOR'TAKEIT'
2098                                    ;TKDIMS= OPTMP2                 ;POINTER TO NEXT DIM FOR 'TAKEIT'
2099
2100                                    ;'TAKER' RECURSES TO DO THE ACTUAL TAKING FOR 'TAKE' AND 'DROP' OP'S
2101                                    ;USING THE FRAMES CREATED BY 'TAKEIT' AND CREATING SP FRAMES AS IT
2102                                    ;RECURSES.  THE SP FRAMES CONSIST OF:
2103                                    ;
2104                                    ;       COUNT FOR THIS LEVEL (GOES TO ZERO)
2105                                    ;       POINTER TO 'TAKEIT' FRAME TO USE
2106                                    ;       RETURN POINTER
2107                                    ;
2108
2109 001536  013703  005072'    TAKER:  MOV     TKDIMS,R3               ;GET # OF DIMS
2110 001542  005713                     TST     (R3)                    ;HOW MANY DIMS?
2111 001544  001447                     BEQ     TAKER2                  ;NONE, FINISHED W/O STARTING
2112 001546  016603  000002    TAKER1: MOV     2(SP),R3                ;GET DIM FRAME POINTER
2113 001552  012302                     MOV     (R3)+,R2                ;RESULT SKIP BEFORE COPY?
2114 001554  100001                     BPL     1$                      ;NO
2115 001556  160200                     SUB     R2,R0                   ;SUBTRACT NEGATIVE NUMBER FOR SKIP
2116 001560  012302             1$:     MOV     (R3)+,R2                ;SOURCE SKIP BEFORE COPY?
2117 001562  100001                     BPL     2$                      ;NO
2118 001564  160204                     SUB     R2,R4                   ;SKIP SOURCE
2119 001566  005713             2$:     TST     (R3)                    ;END OF FRAMES?
2120 001570  001016                     BNE     6$                      ;NO
2121                                    ;
2122                                    ;END-OF-FRAMES ..... DO ACTUAL COPYING
2123                                    ;
2124 001572  016601  000004            MOV     4(SP),R1                ;GET ELEMENTS TO COPY
2125 001576  013702  003212'            MOV     RARGTY,R2               ;GET ARG TYPE
2126 001602                             ASH     SIZTAB(R2),R1           ;CONVERT TO BYTE COUNT
2127
2128 001616  112420             3$:     MOVB    (R4)+,(R0)+             ;COPY CHR
2129 001620                             SOB     R1,3$                   ;DO ALL THE CHRS WANTED
2130 001624  000417                     BR      TAKER2                  ;FINISHED
2131                                    ;
2132                                    ;RECURE TO NEXT LOWER LEVEL
2133                                    ;
2134 001626                     6$:     PUSH    <(R3)+,R3>              ;CREATE FRAME
2135 001632                             CALL    TAKER1                  ;RECURSES! FOILED AGAIN!
2136 001636  016603  000002            MOV     2(SP),R3                ;GET FRAME POINTER
2137 001642  012302                     MOV     (R3)+,R2                ;RESULT SKIP AFTER COPY?
2138 001644  100401                     BMI     7$                      ;NO
2139 001646  060200                     ADD     R2,R0                   ;SKIP RESULT
2140 001650  012302             7$:     MOV     (R3)+,R2                ;SOURCE SKIP AFTER COPY?
2141 001652  100401                     BMI     8$                      ;NO
2142 001654  060204                     ADD     R2,R4                   ;SKIP SOURCE
2143 001656  005366  000004     8$:     DEC     4(SP)                   ;DEC  REPEAT COUNTER TO SEE IF DONE
2144 001662  001331                     BNE     TAKER1                  ;NOT DONE, DO AGAIN
2145 001664                     TAKER2: POP     R2                      ;YES, GET RETURN TO PREVIOUS LEVEL
2146 001666  022626                     CMP     (SP)+,(SP)+             ;CLEAN ARGS OFF STACK
2147 001670  000112                     JMP     (R2)                    ;RETURN UPWARD
2149 001672                             LIST
```

```
   2151 001672                          MOPINS  1REV,146
   2160 001672                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 001672                          MOPINS  REVS,144,T.DIM
   2204 001672                                  LIST
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 58
MIXED FUNCTION          TAKE

   2206 001672                          DOPINS  1ROT,146
   2214 001672                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 001672                          DOPINS  ROT,144,T.DIM
   2328 001672                                  LIST
```

```
2330 001672                          MOPINS  TRAN,126
2415 001672                                  LIST
```

```
   2417 001672                          DOPINS  TRAN,126
   2654 001672                                  LIST
```

```
                                        DOPINS  ENCD,76
                                        .SBTTL  MIXED FUNCTION          ENCODE
                                        ;
                                        ;RETURNS REPRESENTATION OF RIGHT ARG IN NUMBER SYSTEM OF LEFT ARG.
                                        ;THE ROUTINE USES SO MANY POINERS AND CONSTANTS BECAUSE OF THE UGLY WAY
                                        ;ENCODE EXTENDS TO ARRAYS OF HIGH ORDER.  ESSENTIALLY THE EVALUATION HAS
                                        ;TO BE DONE BACKWARDS FROM BOTTOM TO TOP,  AND THREE NESTED LOOPS ARE
                                        ;REQUIRED TO PRODUCE THE RESULT.
                                        ;
                                        ;CONSTANT ASSIGNMENTS:
                                        ;
                                        ;OPTMP  (N)      USE
                                        ;
                                        ;CCNT   (2)      INNER LOOP COUNT (# OF ROWS IN LEFT ARG)
                                        ;RBASE  (3)      PRIMARY RESULT BASE POINTER
                                        ;OPTMP5          INNER LOOP RESULT BASE INCREMENT (-VOLUME OF RIGHT ARRAY)
                                        ;RINC   (7)      RESULT POINTER INCREMENT
                                        ;LCOUNT (8)      OUTER LOOP COUNT (VOLUME OF RIGHT ARG)
                                        ;RCOUNT (9)      INNER LOOP BASE COUNT
                                        ;RCOUNR (10)     INNER LOOP BASE COUNT RESET
                                        ;RARGPT          POINTER TO CURRENT RIGHT ARG
                                        ;LARGPT          PRIMARY LEFT ARG BASE
                                        ;OPTMP6          CURRENT LEFT BASE
                                        ;RESPTR          CURRENT RESULT BASE
                                        ;
                                        ;UGH!

                                        ;
                                        ;CHECK ARGUMENT CONFORMITY
                                        ;
001672  013702  003162'                         MOV     LARGBL,R2       ;POINT TO LEFT ARG
001676  013700  003170'                         MOV     LARGTY,R0       ;GET LEFT ARG TYPE
001702  001001                                  BNE     2$              ;YES, FINE

001704  104417                         1$:     DOMERR                  ;BAD ARGS

001706  013703  003204'                2$:     MOV     RARGBL,R3       ;POINT TO RIGHT ARG
001712  013701  003212'                         MOV     RARGTY,R1       ;GET RIGHT ARG TYPE
001716  001772                                  BEQ     1$              ;OOPS
001720  016137  000000G 003220'                MOV     FRARGC(R1),RARGCV ;The Fetch Routine
001726  016037  000000G 003176'                MOV     FLARGC(R0),LARGCV ;''The Other'' Fetch Routine
001734  020100                                  CMP     R1,R0           ;Which Type is Bigger?
001736  003401                                  BLE     22$             ;The Left > The Right
001740  010100                                  MOV     R1,R0           ;The Right > The Left
001742  010037  003144'                22$:     MOV     R0,RESTYP       ;Save Result Type
001746  016037  000000G 003152'                MOV     FRESCT(R0),RESCVT ;Result Conversion Routine
                                        ;
                                        ;CALCULATE SIZE OF RESULT AND ALLOCATE RESULT BLOCK
                                        ;
001754  016301  000004                          MOV     A.ALEN(R3),R1   ;GET VOLUME OF RIGHT ARG
001760  010137  005106'                         MOV     R1,LCOUNT       ;SAVE AS OUTER LOOP COUNT
001764                                          MUL     A.ALEN(R2),R1   ;TOTAL VOLUME OF RESULT
002000  101741                                  BLOS    1$              ;TOO BIG OR TOO SMALL
002002  116304  000002                          MOVB    A.DIMS(R3),R4   ;# OF DIMS ON LEFT
002006  116200  000002                          MOVB    A.DIMS(R2),R0   ;# OF DIMS ON RIGHT
002012                                          PUSH    <R4,R0,R1>      ;RIGHT DIMS, LEFT DIMS, RESULT ALEN
002020  060400                                  ADD     R4,R0           ;# OF DIMS IN RESULT
```

```
2713 002022                                     PUSH    R0              ;SAVE TO FILL IN HEADER
2714 002024                                     CONINC  R0              ;;INCR FOR SCALAR
2715 002030  006300                             ASL     R0              ;CONVERT TO WORD COUNT
2716 002032  062700  000006                     ADD     #A.DIM1,R0      ;ADD OVERHEAD
2717 002036                                     PUSH    R0              ;SAVE OFFSET TO DATA
2718 002040  013704  003144'                    MOV     RESTYP,R4       ;Get Result Type
2719 002044                                     ASH     SIZTAB(R4),R1   ;Multiply Time
2720 002060  060100                             ADD     R1,R0           ;TOTAL SIZE OF RESULT BLOCK
2721                                    ;
2722                                    ;FILL IN RESULT HEADER
2723                                    ;
2724 002062                                     CALL    GETFRA          ;Get result, SI frame for it
2725 002066  062620                             ADD     (SP)+,(R0)+     ;Add Offset to Data
2726 002070  112620                             MOVB    (SP)+,(R0)+     ;# OF DIMS
2727 002072  113710  003144'                    MOVB    RESTYP,(R0)     ;RESULT IS ALWAYS NUMERIC
2728 002076  152720  000200                     BISB    #A.ASB,(R0)+    ;Flag as Array
2729 002102  012620                             MOV     (SP)+,(R0)+     ;ALEN OF RESULT
2730 002104                                     POP     R4              ;# OF DIMS OF LEFT ARG
2731 002106  010201                             MOV     R2,R1           ;COPY POINTER TO LEFT ARG
2732 002110  062701  000006                     ADD     #A.DIM1,R1      ;POINT TO FIRST DIM
2733 002114  005704                             TST     R4              ;IS LEFT ARG A SCALAR?
2734 002116  001403                             BEQ     10$             ;;YES, SO SKIP LOOP
2735
2736 002120  012120                     3$:     MOV     (R1)+,(R0)+     ;COPY LEFT ARG'S DIM LIST
2737 002122                                     SOB     R4,3$
2738
2739 002126  014137  005072'            10$:    MOV     -(R1),CCNT      ;LAST DIM ON LEFT IS IMPORTANT CONSTANT!
2740 002132                                     POP     R4              ;# OF DIMS OF RIGHT ARG
2741 002134                                     CONINC  R4              ;Cheat For Scalar
2742 002140  010301                             MOV     R3,R1           ;POINT TO RIGHT ARG
2743 002142  062701  000006                     ADD     #A.DIM1,R1      ;POINT TO RIGHT ARG'S DIM LIST
2744
2745 002146  012120                     4$:     MOV     (R1)+,(R0)+     ;COPY RIGHT ARG'S DIM LIST
2746 002150                                     SOB     R4,4$
2747                                    ;
2748                                    ;INITIALIZE POINTERS AND COUNTS
2749                                    ;
2750 002154  016501  177776                     MOV     SI.VAL(SI),R1   ;POINT TO RESULT
2751 002160  016100  000004                     MOV     A.ALEN(R1),R0   ;GET ITS LENGTH
2752 002164  011101                             MOV     (R1),R1         ;MAKE POINTER TO RESULT ABSOLUTE
2753 002166  013704  003144'                    MOV     RESTYP,R4       ;Get Result Type
2754 002172                                     ASH     SIZTAB(R4),R0   ;  Times Element Size (In Bytes)
2755 002206  060001                             ADD     R0,R1           ;POINT PAST END OF RESULT
2756 002210  010137  005074'                    MOV     R1,RBASE        ;TO GET RESULT BASE POINTER
2757
2758 002214  016300  000004                     MOV     A.ALEN(R3),R0   ;GET LENGTH OF RIGHT ARG
2759 002220  013704  003212'                    MOV     RARGTY,R4       ;Get Right Arg Type
2760 002224  005300                             DEC     R0              ;Offset for Last Element
2761 002226                                     ASH     SIZTAB(R4),R0   ;Get Right Arg Length
2762 002242  060037  003206'                    ADD     R0,RARGPT       ;POINT PAST END OF RIGHT ARG
2763
2764 002246  016201  000004                     MOV     A.ALEN(R2),R1   ;GET LENGTH OF LEFT ARG
2765 002252  005301                             DEC     R1              ;MINUS 1 TO POINT AT LAST ELEMENT!
2766 002254  013704  003170'                    MOV     LARGTY,R4       ;Get Left Arg Type
2767 002260                                     ASH     SIZTAB(R4),R1   ;CONVERT
2768 002274  060137  003164'                    ADD     R1,LARGPT       ;POINT PAST END OF LEFT ARG
2769
```

```
2770 002300  016201  000004                    MOV     A.ALEN(R2),R1   ;GET LEFT ALEN
2771 002304  020162  000006                    CMP     R1,A.DIM1(R2)   ;VECTOR OR SCALAR?
2772 002310  001003                            BNE     6$              ;NOPE
2773 002312  010137  005072'                   MOV     R1,CCNT         ;YES, SO CCNT IS LENGTH OF VECTOR
2774 002316  000411                            BR      7$              ;CONTINUE
2775 002320  005000                    6$:     CLR     R0              ;CLEAR FOR DIVIDE
2776 002322                                    DIV     CCNT,R0         ;DIVIDE ALEN BY LAST DIM
2777 002336  010037  005072'                   MOV     R0,CCNT         ;TO GET CCNT
2778
2779 002342  016201  000004            7$:     MOV     A.ALEN(R2),R1   ;GET LENGTH OF LEFT ARG AGAIN
2780 002346  005000                            CLR     R0              ;CLEAR FOR DIVIDE
2781 002350                                    DIV     CCNT,R0         ;DIVIDE BY CCNT
2782 002364  010037  005112'                   MOV     R0,RCOUNR       ;TO GET COUNT RESET
2783 002370  005400                            NEG     R0              ;OFFSETS NEGATIVE (MOVE BACKWARDS)
2784 002372                                    PUSH    R0              ;Save # of Elements
2785 002374  013704  003170'                   MOV     LARGTY,R4       ;Get Left Arg Type
2786 002400                                    ASH     SIZTAB(R4),R0   ;CONVERT FOR FLOATING POINT DATA
2787 002414  010037  003160'                   MOV     R0,LARGFL       ;TO GET LEFT ARG INCREMENT
2788 002420                                    POP     R0              ;Restore The Count
2789 002422                                    MUL     A.ALEN(R3),R0   ;MULTIPLY BY VOLUME OF RIGHT ARG
2790 002436  005301                            DEC     R1              ;OFFSET FOR AUTO-INCREMENT
2791 002440  013704  003144'                   MOV     RESTYP,R4       ;GET RESULT TYPE
2792 002444                                    ASH     SIZTAB(R4),R1   ;DO A SHIFT
2793 002460  010137  005104'                   MOV     R1,RINC         ;TO GET RESULT INCREMENT
2794 002464  016300  000004                    MOV     A.ALEN(R3),R0   ;GET RIGHT ARG'S VOLUME
2795 002470  013704  003144'                   MOV     RESTYP,R4       ;Get The Result Type
2796 002474                                    ASH     SIZTAB(R4),R0   ;CONVERT
2797 002510  005400                            NEG     R0              ;OFFSET IS NEGATIVE
2798 002512  010037  005100'                   MOV     R0,OPTMP5       ;AND SAVE AS BASE INCREMENT
2799                                   ;MAIN LOOPS OF ENCODE PROCESS.  FIRST SET UP COMMON CONSTANTS
2800                                   ;
2801 002516                            DOENCD: LDF     #^F1.0,AC0      ;GET 1.0
2802 002526                                    STF     AC0,AC4         ;Put it in AC4
2803 002534  013703  003206'                   MOV     RARGPT,R3       ;POINT TO RIGHT ARG
2804 002540  005437  003202'                   NEG     RARGFL          ;   is Back Two
2805                                   ;
2806                                   ;RESET FOR NEXT RIGHT ARGUMENT
2807                                   ;
2808 002544  013700  003144'           1$:     MOV     RESTYP,R0       ;Get Result Type
2809 002550  166037  000054'  005074'          SUB     LENTAB(R0),RBASE;BACK UP PRIMARY RESULT BASE
2810 002556  013737  003164'  005102'          MOV     LARGPT,OPTMP6   ;RESET LEFT BASE
2811 002564  013737  005074'  003140'          MOV     RBASE,RESPTR    ;RESET RIGHT BASE
2812 002572  013737  005112'  005110'          MOV     RCOUNR,RCOUNT   ;RESET RESULT BASE COUNTER
2813 002600                                    CALL    @RARGCV         ;Fetch Right Arg
2814 002604                                    STF     AC2,AC5         ;SAVE THE ARG
2815                                   ;
2816                                   ;RESET FOR NEXT LEFT GROUP
2817                                   ;
2818 002612                            2$:     LDF     AC5,AC2         ;GET NEW RIGHT VALUE
2819 002620  013702  005102'                   MOV     OPTMP6,R2       ;LEFT ARG POINTER
2820 002624  013701  003140'                   MOV     RESPTR,R1       ;RESULT POINTER
2821 002630  013700  005072'                   MOV     CCNT,R0         ;COMBINE COUNT
2822                                   ;
2823                                   ;ENCODE CURRENT ARGUMENT WITH RESPECT TO CURRENT LEFT GROUP
2824                                   ;
2825 002634                            3$:     CALL    @LARGCV         ;Get The Next Left Arg
2826 002640                                    STF     AC2,AC0         ;RETURN REMAINDER OF ARG
```

```
2827 002646                                     TSTF    AC1             ;Check For Zero
2828 002654                                     CFCC
2829 002660  001004                             BNE     4$              ;WHICH IS A SPECIAL CASE
2830 002662                                     CLRF    AC2             ;AND ZAP NUMBER
2831 002670  000413                             BR      5$              ;WHICH WOULD NORMALLY BE TRUNCATED
2832                                    ;
2833                                    ;GET NEXT REPRESENTATION
2834                                    ;
2835 002672                             4$:     CALL    D$$RES          ;       R = A ! B
2836 002676                                     SUBF    AC0,AC2         ;       B = B - R
2837 002704                                     CFCC                    ;       IF B=0 THEN USE NEXT RIGHT ARG
2838 002710  001403                             BEQ     5$
2839 002712                                     DIVF    AC1,AC2         ;       B = B % A
2840 002720                             5$:     CALL    @RESCVT         ;Save Result
2841 002724  104402                             SYSERR                  ;***NO NO NO***
2842 002726  063701  005104'                    ADD     RINC,R1         ;POINT TO NEXT SLOT IN RESULT
2843 002732  005300                             DEC     R0              ;SUBTRACT 1
2844 002734  001337                             BNE     3$              ;AND PROCESS REST OF LEFT GROUP
2845
2846 002736  063737  005100' 003140'            ADD     OPTMP5,RESPTR   ;INCREMENT RESULT BASE
2847 002744  013700  003170'                    MOV     LARGTY,R0       ;Get Left Arg Type
2848 002750  166037  000054' 005102'            SUB     LENTAB(R0),OPTMP6 ;MOVE LEFT BASE BACK 1 POSITION
2849 002756  005337  005110'                    DEC     RCOUNT          ;COUNT THIS PASS
2850 002762  003313                             BGT     2$              ;MORE LEFT GROUPS TO PROCESS
2851
2852 002764  005337  005106'                    DEC     LCOUNT          ;MOVE TO NEXT RIGHT ARG
2853 002770  003265                             BGT     1$              ;IF THERE ARE ANY
2854 002772                                     RETURN
2856 002774                                     LIST
```

```
                                DOPINS  DECD,64
                                .SBTTL  MIXED FUNCTION          DECODE

                                ;++
                                ;RESULT IS RIGHT ARG EVALUATED IN NUMBER SYSTEM OF LEFT ARG.
                                ;PARSE FRAMES ARE PLACED ON THE SI STACK AND THE LEFT ARG
                                ;IS CONVERTED TO 'WEIGHT VECTORS' SO THAT THE INNER PRODUCT
                                ;ROUTINE CAN BE USED TO PRODUCE THE RESULT.
                                ;
                                ;MAKE SURE ARGUMENTS ARE NUMERIC
                                ;--
002774  005737  003170'                TST     LARGTY          ;Is Left Arg Numeric?
003000  001001                          BNE     2$              ;YES

003002  104417          1$:     DOMERR                          ;BAD ARGS

003004  005737  003212'  2$:     TST     RARGTY          ;Point to Right Arg
003010  001774                          BEQ     1$              ;NO
                                ;+
                                ;CREATE PARSE FRAMES ON THE SI STACK TO FOOL THE INNER
                                ;PRODUCT ROUTINE WHO IS GOING TO DO MOST OF THE WORK.
                                ;-
003012                          SIFRAM  RARGBL,#T.NUM   ;Right Arg (But not a Val!)
003026                          SIFRAM  #OP.MUL,#T.OPFN+T.SOP
                                                        ;Multiply Operation (A+.*B)
003042                          SIFRAM  #OP.PLU,#T.OPFN+T.SOP
                                                        ;Addition Operator
                                ;+
                                ;GET NEW ARRAY TO HOLD WEIGHTS FOR INNER PRODUCT.  (A NEW PARSE
                                ;FRAME IS PLACED ON THE SI STACK.
                                ;-
003056  013700  003162'          MOV     LARGBL,R0       ;POINT TO LEFT ARG
003062  116002  000002          MOVB    A.DIMS(R0),R2   ;GET NUMBER OF DIMS
003066                          CONINC  R2              ;CONVERT DIMS FOR SCALAR
003072  006302                  ASL     R2              ;CONVERT TO WORDS
003074  062702  000006          ADD     #A.DIM1,R2      ;ADD HEADER SIZE
003100  060002                  ADD     R0,R2           ;TO POINT PAST LAST DIM
003102  022742  000001          CMP     #1,-(R2)        ;IS LAST DIM 1? (SPECIAL CASE)
003106  001070                  BNE     3$              ;NOPE
                                ;+
                                ;SET UP EXPANDED ARG ARRAY
                                ;-
003110  005037  003160'          CLR     LARGFL          ;INCREMENT IS ZERO
003114  013702  003204'          MOV     RARGBL,R2       ;POINT TO RIGHT ARG
003120  016003  000004          MOV     A.ALEN(R0),R3   ;GET SIZE OF LEFT ARG
003124  016246  000006          MOV     A.DIM1(R2),-(SP);PUSH FISRT DIM OF RIGHT ARG
003130                          MUL     @SP,R3          ;AND INCREASE SIZE OF RESULT
003142  103717                  BCS     1$              ;TOO BIG
003144                          PUSH    R3              ;SAVE SIZE
003146                          ASH     PRECSN,R3       ;CONVERT TO FLOATING DATA SIZE
003162  102707                  BVS     1$              ;TOO BIG
003164  116002  000002          MOVB    A.DIMS(R0),R2   ;GET NUMBER OF DIMS IN LEFT ARG
003170                          PUSH    R2              ;SAVE # OF DIMS
003172                          CONINC  R2              ;CONVERT FOR SCALAR
003176  006302                  ASL     R2              ;CONVERT TO WORDS
003200  062702  000006          ADD     #A.DIM1,R2      ;ADD HEADER OFFSET
003204                          PUSH    R2              ;SAVE HEADER SIZE
```

```
2915 003206  010200                          MOV     R2,R0           ;COPY HEADER SIZE
2916 003210  060300                          ADD     R3,R0           ;ADD DATA LENGTH
2917 003212                                  CALL    GETFRA          ;Get result, SI frame for it
2918 003216  062620                          ADD     (SP)+,(R0)+     ;HEADER SIZE
2919 003220                                  POP     R2              ;# OF DIMS
2920 003222  110220                          MOVB    R2,(R0)+        ;SAME # OF DIMS
2921 003224                                  CONINC  R2              ;ADJUST FOR SCALAR
2922 003230  112720  000206                  MOVB    #A.ASB+.FLOAT,(R0)+ ;WEIGHT IS NUMERIC
2923 003234  012620                          MOV     (SP)+,(R0)+     ;ALEN
2924 003236  013701  003162'                 MOV     LARGBL,R1       ;POINT BACK TO LEFT ARG
2925 003242  062701  000006                  ADD     #A.DIM1,R1      ;POINT TO FIRST DIM
2926
2927 003246  012120                  5$:     MOV     (R1)+,(R0)+     ;COPY DIM LIST
2928 003250                                  SOB     R2,5$           ;ONCE PER DIM
2929 003254  012660  177776                  MOV     (SP)+,-2(R0)    ;BUT LAST DIM GOT BIGGER FOR CONFORMITY
2930 003260  010001                          MOV     R0,R1           ;DATA POINTER
2931 003262  016500  177776                  MOV     SI.VAL(SI),R0   ;ARRAY POINTER
2932 003266  000411                          BR      DECD1           ;AND CREATE WEIGHTS
2933                                 ;+
2934                                 ;GET WEIGHT ARRAY OF SAME SHAPE AS LEFT ARG
2935                                 ;-
2936 003270  013737  003174' 003160' 3$:     MOV     LARGLE,LARGFL   ;GET THE LEFT ARG LENGTH
2937 003276  005437  003160'                 NEG     LARGFL          ;MAKE IT A DECREMENT
2938 003302  012701  000006                  MOV     #.FLOAT,R1      ;RESULT IS FLOAT
2939 003306                                  CALL    GETSM2          ;GET SAME SHAPED EMPTY ARRAY
2940
2941 003312  116002  000002          DECD1:  MOVB    A.DIMS(R0),R2   ;GET NUMBER OF DIMS IN LEFT ARG
2942 003316                                  CONINC  R2              ;FIXUP SCALAR
2943 003322  006302                          ASL     R2              ;CONVERT TO WORDS
2944 003324  060002                          ADD     R0,R2           ;POINT PAST LAST DIM
2945 003326  062702  000006                  ADD     #A.DIM1,R2      ;MOVE POINTER PAST HEADER TO DIM LIST
2946 003332  014246                          MOV     -(R2),-(SP)     ;LAST DIM GOES ON STACK
2947 003334  005002                          CLR     R2              ;CLEAR FOR DIVIDE
2948 003336  016003  000004                  MOV     A.ALEN(R0),R3   ;VOLUME OF ARRAY
2949 003342  001477                          BEQ     1$              ;EMPTY ARRAYS NOT LEGAL
2950 003344                                  DIV     @SP,R2          ;NUMBER OF ROWS
2951 003356                                  PUSH    R2              ;SAVE ROW COUNT
2952 003360  016002  000004                  MOV     A.ALEN(R0),R2   ;GET VOLUME AGAIN
2953 003364                                  ASH     PRECSN,R2       ;CONVERT FOR FLOATING POINT DATA
2954 003400  102460                          BVS     1$              ;TOO BIG
2955 003402  060201                          ADD     R2,R1           ;POINT PAST END OF RESULT
2956 003404  013703  003162'                 MOV     LARGBL,R3       ;POINT TO LEFT ARG
2957 003410  016302  000004                  MOV     A.ALEN(R3),R2   ;GET ITS LENGTH
2958 003414  005302                          DEC     R2              ;MINUS 1 SO WE POINT TO END
2959 003416                                  ASH     LARGSF,R2       ;MULTIPLY BY ELEMENT SIZE
2960 003432  102443                          BVS     1$              ;TOO BIG
2961 003434  063702  003164'                 ADD     LARGPT,R2       ;AND POINT TO LAST ELEMENT ON LEFT
2962 003440                                  POP     R4              ;# OF ROWS
2963 003442  013700  003170'                 MOV     LARGTY,R0       ;Get Left Arg Type
2964 003446  016037  000054' 005070'         MOV     LENTAB(R0),OPTMP1 ;Size For Decrement
2965 003454  016037  000000G 003176'         MOV     FLARGC(R0),LARGCV ;Fetch Routine
2966 003462  011600                          MOV     (SP),R0         ;Get the Minor Loop Count
2967                                 ;+
2968                                 ;SET UP WEIGHTS
2969                                 ;-
2970 003464                          10$:    LDF     #^F1.0,AC0      ;WEIGHTS START WITH 1
2971 003474  000405                          BR      13$             ;AND LEFTMOST ARG IS IGNORED
```

```
2972
2973 003476                          12$:    CALL    @LARGCV         ;GET LEFT Arg
2974 003502                                  MULF    AC1,AC0         ;Multiply
2975 003510                          13$:    STF     AC0,-(R1)       ;FILL IN RESULT (BACKWARDS)
2976 003516                                  SOB     R0,12$          ;PROCESS ENTIRE ROW
2977 003522  163702  005070'                SUB     OPTMP1,R2       ;Back Up 1 Element
2978 003526  011600                          MOV     (SP),R0         ;RESET ROW LENGTH COUNTER
2979 003530                                  SOB     R4,10$          ;AND PROCESS NEXT ROW
2980 003534  005726                          TST     (SP)+           ;POP ROW LENGTH
2981                                 ;+
2982                                 ;DO INNER PRODUCT TO OBTAIN RESULT
2983                                 ;-
2984 003536                                  CALLR   D$INNR
2985
2986 003542  104417                  1$:     DOMERR                  ;TOO BIG!
2988 003544                                  LIST
```

```
 2990 003544                          MOPINS  GDWN,136,T.DIM
 3001 003544                                  LIST
 3002
 3003
 3004 003544                          MOPINS  GDUP,140,T.DIM
 3219 003544                                  LIST
```

E 9

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 64
MIXED FUNCTION          DECODE

   3221 003544                          MOPINS  MMD,122
   3316 003544                                  LIST
```

F 9

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 65
MIXED FUNCTION          DECODE

   3318 003544                          DOPINS  DMD,122
```

G 9

```
2  MACRO M1113  08-FEB-80 09:16  PAGE 66
 FUNCTION          DECODE

 4 003544                                  LIST
```

H 9

```
3866 003544                                DOPINS  MEM,70
3867                                       .SBTTL  MIXED FUNCTION          MEMBERSHIP
3868
3869 003544                                        CALL    GETREG          ;GET SOME REGISTERS AND BITS
3870 003550  016437  000004  005074'              MOV     A.ALEN(R4),OPTMP3 ;RIGHT ARG SIZE
3871 003556  042700  177771                       BIC     #^C<A.NUM>,R0   ;CLEAR OUT GARBAGE
3872 003562  010002                              MOV     R0,R2           ;ASSUME LEFT TYPE
3873 003564  001006                              BNE     2$              ;IT IS NUMERIC
3874 003566  042701  177771                       BIC     #^C<A.NUM>,R1   ;CLEAR OUT GARBAGE
3875 003572  001411                              BEQ     3$              ;TYPES MATCH
3876 003574  005037  005074'          1$:       CLR     OPTMP3          ;COUNT IS ZERO, 'CAUSE TYPES DON'T MATCH
3877 003600  000423                              BR      4$              ;AND GO
3878
3879 003602  042701  177771          2$:       BIC     #^C<A.NUM>,R1   ;CLEAR GARBAGE BITS
3880 003606  001772                              BEQ     1$              ;NOT SAME, ERROR
3881 003610  020201                              CMP     R2,R1           ;WHICH TYPE BIGGER?
3882 003612  002001                              BGE     3$              ;EVIDENTLY, THE 1ST
3883 003614  010102                              MOV     R1,R2           ;NO, IT'S THE 2ND
3884 003616  016237  000000G 005122' 3$:        MOV     COMCT(R2),COMPTR;COMPARE ROUTINE
3885 003624  006302                              ASL     R2              ;TIMES 2
3886 003626  006302                              ASL     R2              ;  "   4
3887 003630  060200                              ADD     R2,R0           ;LEFT ARG FETCH ROUTINE
3888 003632  016037  000000G 003176'             MOV     LARGCT(R0),LARGCV ;SO HERE IT IS
3889 003640  060201                              ADD     R2,R1           ;RIGHT ARG FETCH ROUTINE
3890 003642  016137  000000G 003220'             MOV     RARGCT(R1),RARGCV ;SO, HERE IT IS
3891 003650  010300                    4$:       MOV     R3,R0           ;COPY POINTER TO LEFT ARG
3892 003652  012701  000002                      MOV     #.BOOLEAN,R1    ;RESULT TYPE IS BOOLEAN
3893 003656                                      CALL    GETSM2          ;GET SAME SIZE ARRAY
3894 003662  011001                              MOV     @R0,R1          ;POINT TO FIRST ELEMENT OF RESULT
3895 003664  011302                              MOV     @R3,R2          ;POINT TO FIRST ELEMENT OF LEFT ARG
3896 003666  016004  000004                      MOV     A.ALEN(R0),R4   ;GET NUMBER OF ELEMENTS IN RESULT
3897 003672  001424                              BEQ     9$              ;NOTHING THERE
3898                                       ;
3899                                       ;PROCESS NEXT ARG IN LEFT ARRAY
3900                                       ;
3901 003674  013700  005074'          5$:       MOV     OPTMP3,R0       ;GET COUNT FOR INNER LOOP
3902 003700  001416                              BEQ     8$              ;NOTHING THERE, EVIDENTLY
3903 003702                                      CALL    @LARGCV         ;FETCH LEFT ARG
3904 003706  013703  003206'                     MOV     RARGPT,R3       ;POINT TO FIRST ELEMENT IN RIGHT ARG
3905                                       ;
3906                                       ; INNER LOOP - DO COMPARES
3907                                       ;
3908 003712                              6$:       CALL    @RARGCV         ;GET RIGHT ARG
3909 003716                                        CALL    @COMPTR         ;COMPARE WITH LEFT ARG
3910 003722  001403                              BEQ     7$              ;IF EQUAL, STOP LOOP
3911 003724                                      SOB     R0,6$           ;LOOP FOR EACH ELEMENT
3912 003730  000402                              BR      8$              ;NOT FOUND, PUT A ZERO IN THE RESULT
3913                                       ;
3914                                       ;MARK A HIT IN THE RESULT ARRAY
3915                                       ;
3916 003732  012700  000001          7$:       MOV     #1,R0           ;FOUND, PUT A ONE INTO THE RESULT
3917 003736  110021                    8$:       MOVB    R0,(R1)+        ;STORE THE RESULT
3918 003740                                      SOB     R4,5$           ;LOOP FOR ALL
3919 003744                              9$:       RETURN
3921 003746                                        LIST
```

```
 3923 003746                            MOPINS  NUB,106
 3986 003746                                    LIST
 3987
 3988
 3989
 3990
 3991
 3992 003746                            DOPINS  UNON,106
 4022 003746                                    LIST
 4023
 4024
 4025
 4026
 4027
 4028 003746                            DOPINS  INTR,66
 4042 003746                                    LIST
 4043
 4044
 4045
 4046
 4047
 4048 003746                            DOPINS  SDIF,50
 4161 003746                                    LIST
```

J 9


```
4163 003746                          DOPINS  SCON,234
4179 003746                                  LIST
4180
4181
4182 003746                          DOPINS  CONT,112
4200 003746                                  LIST
4201
4202
4203 003746                          DOPINS  PSUB,236
4209 003746                                  LIST
4210
4211
4212 003746                          DOPINS  SUBS,116
4288 003746                                  LIST
```

K 9

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 70
MIXED FUNCTION          MEMBERSHIP

   4290 003746                          MOPINS  EXE,70
   4336 003746                                  LIST
```

```
 4338 003746                            MOPINS  ENCD,76
 4411 003746                                    LIST
```

M 9


```
4413 003746                          MOPINS  FMT,160
4423 003746                                  LIST
4424
4425
4426
4427
4428
4429 003746                          DOPINS  FMT,160
```

N 9

```
   4953 003746                                    LIST
```

```
                                        MOPINS  IBM,132,,,MONIBM
                                                LIST

                                        DOPINS  IBM,132
                                        .SBTTL  MIXED FUNCTION          I BEAM

                                                MOV     #.INTEGER,R1    ;Corvert left arg to integer (if scalar)
                                                CALL    GETSCA          ;Get a scalar, convert
                                                MOV     @(R0),R3        ;Get the value
                                                SUB     #IBMHIH+1,R3    ;TOO SMALL?
                                                ADD     #IBMHIH-IBMLOW+1,R3;OR TOO BIG?
                                                BCC     2$              ;YES
                                                MOV     SI,R2           ;Point to the SI
                                                ADD     #SI.LEN*2,R2    ;Back up by two elements
                                                MOV     #.INTEGER,R1    ;Convert to integer
                                                CALL    CONVER          ;Convert to integer
                                                MOV     (R0),R0         ;Point to self
                                                ASL     R3              ;SHIFT FOR WORD TABLE
                                                ADD     R3,PC           ;Branch to routine

                                                BR      IBMARB          ;ARBOUT FOR NED THANHOUSER OF TEK
                                                BR      IBMPEK          ;PEEK AT CORE
                                                BR      IBMPOK          ;POKE CORE
                                                BR      IBMLOC          ;SET A CORE LOCATION

                                        1$:     DOMERR                  ;DOMAIN ERROR
                                        2$:     NYIERR                  ;NOT YET IMPLEMENTED

                                        IBMLOW=^D20                     ;LOWEST I BEAM ARG
                                        IBMHIH=^D23                     ;HIGHEST I BEAM ARG ALLOWED

                                        ;+
                                        ; (I Beam single character output
                                        ;-
                                        IBMARB: MOV     <2*SI.LEN>+SI.VAL(SI),R4
                                                                        ;POINT TO RIGHT ARG
                                                MOV     A.ALEN(R4),R3   ;Get count to transfer
                                                MOV     (R4),R2         ;COPY POINTER TO FIRST ELEMENT
                                        1$:     MOV     (R2)+,R0        ;GET NEXT ARG
                                                BIC     #UPPER,R0       ;MASK TO 8 BITS
                                                CALL    TTOUT           ;PRINT CHR
                                                SOB     R3,1$           ;LOOP IF ANOTHER CHARACTER
                                                BIC     #F.PRNT,FLAGS   ;DON'T PRINT RESULT
                                                JMP     SAME            ;RETURN THE SAME ARRAY GIVEN


                                        ;+
                                        ; (I Beam) Set core location
                                        ;-
                                        IBMLOC: MOV     (R0),R0         ;Get core address
                                                BIC     #1,R0           ;MAKE SURE IT'S EVEN
                                                MOV     R0,CORPTR       ;SAVE AS THE CORE POINTER
                                                JMP     EMPTY           ;EMPTY RESULT


                                        ;+
```

```
                                        ; (I Beam) Core Peek
                                        ;-
5016 004114  011002             IBMPEK  MOV     (R0),R2         ;Get # of locations to peek
5017 004116  002430                     BLT     2$              ;Must be positive
5018 004120  020227  020004             CMP     R2,#8196.       ;MAXIMUM OF 8K PEEK
5019 004124  101025                     BHI     2$              ;TOO BIG IS TOO BAD
5020 004126  010200                     MOV     R2,R0           ;COPY SIZE
5021 004130  006300                     ASL     R0              ;Count in bytes
5022 004132  062700  000010             ADD     #A.DIM1+2,R0    ;ADD IN HEADER SIZE
5023 004136                             CALL    GETFRA          ;Get result, SI frame for it
5024 004142  062720  000010             ADD     #A.DIM1+2,(R0)+ ;LOAD OFFSET
5025 004146  012720  102001             MOV     #<A.ASB+.INTEGER>*400+1,(R0)+   ;LOAD BITS / RANK
5026 004152  010220                     MOV     R2,(R0)+        ;LOAD SIZE
5027 004154  010220                     MOV     R2,(R0)+        ;FIRT DIM SAME AS SIZE
5028 004156  001407                     BEQ     3$              ;DONE IF NONE TO PEEK AT
5029 004160  013701  005120'            MOV     CORPTR,R1       ;GET CURRENT CORE POINTER
5030 004164  012120             1$:     MOV     (R1)+,(R0)+     ;Copy, integer form
5031 004166                             SOB     R2,1$           ;FILL WHOLE ARRAY
5032 004172  010137  005120'            MOV     R1,CORPTR       ;UPDATE CORE POINTER
5033 004176                     3$:     RETURN                  ;FINIS!
5034
5035 004200  104417             2$:     DOMERR                  ;DOMAIN ERROR
5036
5037                                    ;+
5038                                    ; (I Beam) Poke Core
5039                                    ;-
5040 004202  013701  005120'    IBMPOK: MOV     CORPTR,R1       ;GET CORE POINTER
5041 004206  012021             1$:     MOV     (R0)+,(R1)+     ;Move it to core
5042 004210                             SOB     R3,1$           ;SEE IF MORE
5043 004214  010137  005120'            MOV     R1,CORPTR       ;SAVE CORE POINTER
5044 004220  000137  000000G            JMP     EMPTY           ;EMPTY RESULT
5046 004224                             LIST
```

```
5048 004224                         MOPINS  QUAD,74,T.NIL+T.SPOP
5075 004224                                 LIST
5076
5077
5078
5079
5080
5081 004224                         DOPINS  QUAD,74,T.NIL+T.SPOP
5097 004224                                 LIST
5098
5099
5100
5101
5102
5103 004224                         MOPINS  QQAD,142,T.NIL+T.SPOP
5135 004224                                 LIST
5136
5137
5138
5139
5140
5141 004224                         DOPINS  QQAD,142,T.NIL+T.SPOP
5152 004224                                 LIST
5153
5154
5155
5156
5157
5158 004224                         MOPINS  DQAD,134,T.NIL+T.SPOP
5192 004224                                 LIST
5193
5194
5195
5196 004224                         DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 004224                                 LIST
```

```
5201 004224                             MOPINS  BXAV,162,T.NIL
5219 004224                                     LIST
5220
5221
5222
5223
5224
5225 004224                             MOPINS  BXCT,164,T.NIL+T.SPOP
5234 004224                                     LIST
5235
5236
5237 004224                             DOPINS  BXCT,164,T.NIL+T.SPOP
5272 004224                                     LIST
5273
5274
5275
5276
5277
5278 004224                             MOPINS  BXIO,166,T.NIL+T.SPOP
5287 004224                                     LIST
5288
5289
5290 004224                             DOPINS  BXIO,166,T.NIL+T.SPOP
5302 004224                                     LIST
5303
5304
5305
5306
5307 004224                             MOPINS  BXLC,212,T.NIL
5340 004224                                     LIST
5341
5342
5343
5344
5345
5346
5347 004224                             MOPINS  BXPP,170,T.NIL+T.SPOP
5356 004224                                     LIST
5357
5358
5359 004224                             DOPINS  BXPP,170,T.NIL+T.SPOP
5381 004224                                     LIST
5382
5383
5384
5385
5386
5387
5388 004224                             MOPINS  BXPW,172,T.NIL+T.SPOP
5397 004224                                     LIST
5398
5399
5400 004224                             DOPINS  BXPW,172,T.NIL+T.SPOP
5417 004224                                     LIST
5418
5419
5420
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 76-1
MIXED FUNCTION          I BEAM


   5421
   5422
   5423 004224                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 004224                                  LIST
   5433
   5434
   5435 004224                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 004224                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 004224                          MOPINS  BXTS,200,T.NIL
   5562 004224                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 004224                          MOPINS  BXWA,176,T.NIL
   5592 004224                                  LIST
```



```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 77
```

MIXED FUNCTION I BEAM

```
 5594 004224                          MOPINS  BXCR,206
 5716 004224                                  LIST
```

```
   5718 004224                          MOPINS  BXEX,202
   5775 004224                                  LIST
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 79
MIXED FUNCTION          I BEAM

   5777 004224                          MOPINS  BXFX,210
   5879 004224                                  LIST
```


APL002 MACRO M1113 08-FEB-80 09:16 PAGE 80

J 10

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 80
MIXED FUNCTION          I BEAM

   5881 004224                          MOPINS  BXNC,204
   5950 004224                                  LIST
```

K 10

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 81
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 81
MIXED FUNCTION          I BEAM

   5952 004224                          MOPINS  BXNL,214
   5963 004224                                  LIST
   5964
   5965
   5966 004224                          DOPINS  BXNL,214
   6104 004224                                  LIST
```


APL002 MACRO M1113 08-FEB-80 09:16 PAGE 82

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 82
MIXED FUNCTION          I BEAM

   6106 004224                          DOPINS  FGET,150,T.DIM
   6330 004224                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 004224                          DOPINS  FPUT,152,T.DIM
   6485 004224                                  LIST
```



```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 83
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 83
MIXED FUNCTION          I BEAM

   6487 004224                          DOPINS  FPNT,154,T.DIM
   6714 004224                                  LIST
```

```
   6716 004224                          MOPINS  1SCN,130
   6723 004224                                  LIST
   6724
   6725
   6726 004224                          MOPINS  SCAN,62,T.DIM
   6732 004224                                  LIST
   6733
   6734
   6735 004224                          MOPINS  1RDT,124
   6741 004224                                  LIST
   6742
   6743
   6744 004224                          MOPINS  RDCT,60,T.DIM
   7044 004224                                  LIST
```

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 85
MIXED FUNCTION          I BEAM

    7046 004224                          MOPINS  GETIDN,0
    7193 004224                                  LIST
```



```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 86
```

```
APLOO2  MACRO M1113  08-FEB-80 09:16  PAGE 86
MIXED FUNCTION          I BEAM

   7195 004224                          MOPINS  SATAN,0
   7404 004224                                  LIST
```



```
APLOO2  MACRO M1113  08-FEB-80 09:16  PAGE 87
```

D 11

APLOO2 MACRO M1113 08-FEB-80 09:16 PAGE 87
MIXED FUNCTION I BEAM

```
7406 004224                          MOPINS  OUTR,156
7476 004224                                  LIST
```

E 11

APLOO2 MACRO M1113 08-FEB-80 09:16 PAGE 88

```
 7478 004224                          MOPINS  DOUTER,0
 7504 004224                                  LIST
 7505
 7506
 7507
 7508
 7509
 7510 004224                          MOPINS  PRDSET,0
 7564 004224                                  LIST
 7565
 7566
 7567
 7568
 7569
 7570 004224                          MOPINS  INSPAC,0
 7612 004224                                  LIST
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 89
MIXED FUNCTION          I BEAM

   7614 004224                          DOPINS  INNR,156
```



```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 90
```

APL002 MACRO M1113 08-FEB-80 09:16 PAGE 90
MIXED FUNCTION I BEAM

```
7960 004224                                LIST
```


APL002 MACRO M1113 08-FEB-80 09:16 PAGE 91

APL002 MACRO M1113 08-FEB-80 09:16 PAGE 91
MIXED FUNCTION I BEAM

```
   7962 004224                          MOPINS  SUBR,0
```


APL002 MACRO M1113 08-FEB-80 09:16 PAGE 92

APL002 MACRO M1113 08-FEB-80 09:16 PAGE 92
MIXED FUNCTION I BEAM

```
    8725 004224                                    LIST
```


APL002 MACRO M1113 08-FEB-80 09:16 PAGE 93

J 11

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 93
MIXED FUNCTION          I BEAM

   8727 004224                          MOPINS  RAND,0
   8771 004224                                  LIST
```

K 11

```
APL002  MACRO M1113  08-FEB-80 09:16  PAGE 94
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 94
MIXED FUNCTION          I BEAM

      1         000001                          .END
```

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 94-1
SYMBOL TABLE

AC0     =%000000            CHRFRO= 000135          CTRMAX  000172R     025 D$TAKE  000634RG    020 D.TAKE= 000002
AC1     =%000001            CHRLBR= 000073          C.ALOC= 000162          D$$RES= ****** GX       D.TRAN= 000004
AC2     =%000002            CHRLPR= 000072          C.BLOK= 000064          D$1EXP  000212RG    020 D.UNON= 000006
AC3     =%000003            CHRMNS= 000137          C.BLSZ= 000140          D.AND = 000000          D.$MAX= 000000
AC4     =%000004            CHRMUL= 000075          C.BUFF= 177760          D.BXCT= 000007          D.$MIN= 000000
AC5     =%000005            CHRNEG= 000100          C.BUFL= 177762          D.BXIO= 000007          D.$MUL= 000000
ALLINT  003274R     025     CHRPDL= 000271          C.CCNT= 177766          D.BXNL= 000007          D.$PLU= 000000
ARYRET  003276R     025     CHRPLU= 000055          C.CPTR= 177764          D.BXPP= 000007          D.$RES= 000000
ASCQUO= 000042              CHRPWR= 000120          C.CZ  = 001000          D.BXPW= 000007          D.$SUB= 000000
A.ALEN= 000004              CHRQST= 000121          C.DIG = 000040          D.BXRL= 000007          D.1COM= 000001
A.ARG = 000040              CHRQUO= 000113          C.FASW= 020000          D.CIRC= 000001          D.1EXP= 000002
A.ASB = 000200              CHRRBR= 000047          C.FLEN= 000010          D.COMB= 000001          D.1ROT= 000003
A.BITS= 000003              CHRRPR= 000042          C.FLSH= 004000          D.COMP= 000001          EDICHR  005063R     025
A.CON = 000020              CHRSEM= 000074          C.FRC = 000001          D.CONT= 000006          EDICNT  005064R     025
A.DIMS= 000002              CHRSMO= 000112          C.IDEN= 000010          D.CTLM= 000000          EDIFLG  005062R     025
A.DIM1= 000006              CHRSPA= 000040          C.NAME= 000102          D.DEAL= 000003          EDIT  = 000003
A.NUM = 000006              CHRUPA= 000131          C.NOBF= 000400          D.DECD= 000002          EIS   = 000000
A.PTR = 000000              CHR.BS= 000010          C.NUM = 000020          D.DIV = 000000          ELESIZ  005132R     025
A.TMP = 000010              CHR.CC= 000003          C.OP  = 000004          D.DMD = 000010          EMPTY = ****** GX
BADDIR= 104463              CHR.CR= 000015          C.OPN = 010000          D.DQAD= 177777          ENDCOR  000004R     025
BADERR= 104437              CHR.CZ= 000032          C.RDLN= 177776          D.DROP= 000002          EOF   = 104475
BLARGC= ****** GX           CHR.GT= 000046          C.REWT= 002000          D.ENCD= 000002          ERRMAX= 000166
BLZBUB  005116R     025     CHR.LF= 000012          C.RND = 000010          D.EQ  = 000000          ERRNUM  004772R     025
BRSELF  005046R     025     CHR.LT= 000043          C.SBIT= 177756          D.EXPD= 000002          ERRPTR  004770R     025
CCNT    005072R     025     CHR.SI= 000017          C.SPFX= 000100          D.FGET= 000005          ERR0    004774R     025
CHANEX  003060R     025     CHR.SL= 000057          C.SPGL= 000200          D.FMT = 000004          EXEBUF  005054R     025
CHAN00  000256R     025     CHR.SO= 000016          C.SPLX= 000001          D.FPNT= 000005          EXPND1  000236R     020
CHAN01  000330R     025     CHR.TB= 000011          C.SPRS= 000002          D.FPUT= 000005          E$CT    005056R     025
CHAN02  000512R     025     CHR.UL= 000106          C.SRC = 177774          D.GE  = 000000          E$PT    005060R     025
CHAN03  000674R     025     CHR..A= 000141          C.SRCP= 000024          D.GT  = 000000          FA.APD= 000100
CHAN04  001056R     025     CHR..E= 000145          C.TT  = 000004          D.IBM = 000002          FA.CRE= 000010
CHAN05  001240R     025     CHR..S= 000163          C.WIDC= 177770          D.IND = 000002          FA.DLK= 001000
CHAN06  001422R     025     CHR..T= 000164          C.WIDR= 177772          D.INNR= 000013          FA.ENB= 100000
CHAN07  001604R     025     CHR..U= 000165          DEBUG = 000000          D.INTR= 000006          FA.EXC= 002000
CHAN08  001766R     025     CHR..Z= 000172          DECD1   003312R     020 D.LE  = 000000          FA.EXT= 000004
CHAN09  002150R     025     CHR..0= 000060          DEFERR= 104405          D.LOG = 000001          FA.NSP= 000100
CHAN10  002332R     025     CHR..1= 000061          DEVIL   005114R     025 D.LT  = 000000          FA.NSR= 000200
CHAN11  002514R     025     CH.AND= 000001          DEVNTF= 104470          D.MAX = 000000          FA.POS= 010000
CHAN12  002676R     025     CMDBUF  003250R     025 DFXDIM= ****** GX       D.MEM = 000002          FA.RD = 000001
CHAN13  000022R     023     CMDMAX= 000010          DIGITS  000006R     025 D.MIN = 000000          FA.RWD= 004000
CHAN14  000204R     023     CNTSIZ  005130R     025 DIMAX = 000020          D.MUL = 000000          FA.SEQ= 040000
CHNOPN= 104471              COMBIN  005076R     025 DIMBLK  003112R     025 D.NE  = 000000          FA.SHR= 000040
CHRAND= 000051              COMCNT  005126R     025 DIMPTR  003114R     025 D.NN  = 000000          FA.TMP= 000020
CHRBEL= 000007              COMCT = ****** GX       DIMTYP  003116R     025 D.NR  = 000000          FA.WCK= 020000
CHRBKR= 000133              COMERR= 104425          DIRFUL= 104466          D.OR  = 000000          FA.WRT= 000002
CHRBOL= 000200              COMPTR  005122R     025 DOENCD  002516R     020 D.PLUS= 000000          FD.BLK= 000010
CHRBOX= 000114              COMRTN  003156R     025 DOMERR= 104417          D.PSUB= 000006          FD.CCL= 000002
CHRCMA= 000054              COMSKP  005124R     025 DPTERR= 104416          D.PWR = 000001          FD.COM= 020000
CHRCOL= 000076              CONVER= ****** GX       DYDCNT  003224R     025 D.QQAD= 000000          FD.CR = 000002
CHRDEL= 000107              CORPTR  005120R     025 D$DECD  002774RG    020 D.QUAD= 000000          FD.DIR= 000010
CHRDIV= 000053              CS.ASS= 000200          D$DROP  000502RG    020 D.RES = 000000          FD.FTN= 000001
CHRDLR= 000176              CS.BOO= 000010          D$ENCD  001672RG    020 D.RHO = 000000          FD.F11= 040000
CHRDLT= 000110              CS.FLO= 000020          D$EXPD  000232RG    020 D.ROT = 000003          FD.INS= 000010
CHRDOT= 000056              CS.IDN= 000040          D$IBM   003746RG    020 D.SCON= 000006          FD.ISP= 002000
CHREP = 000105              CS.ONE= 000100          D$IND   000000RG    020 D.SDIF= 000006          FD.MNT= 100000
CHRERR= 104431              CTEPS   000152R     025 D$INNR= ****** GX       D.SUB = 000000          FD.OSP= 004000
CHRFIL= 000177              CTMAX   000212R     025 D$MEM   003544RG    020 D.SUBS= 000006          FD.PLC= 000004
```

APLO02 MACRO M1113 08-FEB-80 09:16 PAGE 94-2
SYMBOL TABLE

```
FD.PRN= 000004          F.BKDS= 000020          F.R3  = 000016          LARGCV  003176R 025 M.CEI = 177777
FD.PSE= 010000          F.BKEF= 000050          F.R4  = 000020          LARGFL  003160R 025 M.CHR = 000202
FD.RAH= 000001          F.BKP1= 000051          F.SEQN= 000100          LARGLE  003174R 025 M.CMT = 000206
FD.RAN= 000002          F.BKST= 000024          F.SPDV= 000072          LARGLN  003166R 025 M.DOUT= 000006
FD.REC= 000001          F.BKVB= 000064          F.SPUN= 000074          LARGPT  003164R 025 M.DQAD= 000006
FD.RWM= 000001          F.BR  = 000020          F.STBK= 000036          LARGSF  003172R 025 M.ENCD= 000004
FD.SDI= 000020          F.CHR = 000075          F.STK = 000022          LARGTY  003170R 025 M.EXE = 000000
FD.SQD= 000040          F.CNTG= 000034          F.STOP= 000004          LARGVA  003200R 025 M.EXP = 000001
FD.TTY= 000004          F.CODE= 000010          F.STP = 000001          LBINC   005100R 025 M.FACT= 000001
FD.WBH= 000002          F.DFNB= 000046          F.SUSP= 000002          LBUFF   003316R 025 M.FIL = 000177
FF.CHR= 000005          F.DSPT= 000044          F.TRC = 000002              LBUFLN= 000621      M.FLR = 177777
FF.NV = 000003          F.DVNM= 000134          F.UNIT= 000136          LCOUNT  005106R 025 M.FMT = 000004
FF.POE= 000002          F.EFBK= 000010          F.URBD= 000020              LENERR= 104412      M.GDUP= 000003
FF.RWD= 000001          F.EFN = 000050          F.VAL = 177776          LENTAB  000054R 025 M.GDWN= 000003
FF.RWF= 000006          F.END = 002000          F.VBN = 000064          LEXPTR  003100R 025 M.GETI= 000006
FF.SPC= 000004          F.EOBB= 000032          F.VBSZ= 000060          LFTOP   005066R 025 M.IBM = 000003
FIXADR  005012R 025     F.ERR = 000052          GETCOR= ****** GX       LINC    005102R 025 M.IDN = 000000
FIXRET  005040R 025     F.FACC= 000043          GETFRA= ****** GX       LINE    005016R 025 M.INSP= 000006
FILNTF= 104467          F.FFBY= 000014          GETREG= ****** GX       LININC  005026R 025 M.IOTA= 000000
FIS   = 000000          F.FLAG= 000002          GETSCA= ****** GX       LISIZE= 000020      M.LIT = 000205
FISCON= 046000          F.FLB = 000100          GETSM2= ****** GX       LITOP   005072R 025 M.LN  = 000001
FIXDIM= ****** GX       F.FNAM= 000110          GNOME   003102R 025 LNLNG = 104541          M.MMD = 000010
FIXPTR  003074R 025     F.FNB = 000102          HASHC   005136R 025 LOCLCT  005036R 025 M.NEG = 000000
FLAGS   004764R 025     F.FNDF= 000010          HASHTB  005140R 025 LOWCOR  000002R 025 M.NOT = 000000
FLARGC= ****** GX       F.FTYP= 000116          HSHMSK= 177601          LSTERP  004766R 025 M.NUB = 000006
FLCBUF  003250R 025     F.FUN = 000010          HSHSIZ= 000200          LWIDTH  000070R 025 M.NUM = 000204
FLINE   005014R 025     F.FVER= 000120          IASH  = ****** GX           L.BITS= 000003      M.OUTR= 000012
FLOEXP  003270R 025     F.HEAD= 000006          IBMARB  004034R 020 L.CNT = 000002          M.PITM= 000000
FLONEG  003272R 025     F.HIBK= 000004          IBMHIH= 000027              L.PRNT= 000001      M.PRDS= 000006
FLSWTB  000222R 025     F.LBL = 000020          IBMLOC  004076R 020 L.PTR = 000000          M.QQAD= 000000
FLTLEN= 000010          F.LINE= 000012          IBMLOW= 000024              L.PTR1= 000004      M.QUAD= 000000
FNINSI  005044R 025     F.LOC = 000004          IBMPEK  004114R 020 L.SUBS= 000002          M.RAND= 000012
FORIGN  000032R 025     F.LSTC= 000004          IBMPOK  004202R 020 MACHIN= 000106          M.RAVL= 000000
FO.APD= 000106          F.LUN = 000042          ICHAN   005050R 025 MAXCHN= 000014          M.RCIP= 000000
FO.MFY= 000002          F.MBCT= 000054          IDIV  = ****** GX       MONBUF  003120R 025 M.RDCT= 000011
FO.RD = 000001          F.MBC1= 000055          ILFNAM= 104464          MONCVT  003132R 025 M.REVS= 000003
FO.UPD= 000006          F.MBFG= 000056          ILLCHN= 104540          MONLEN  003130R 025 M.ROLL= 000000
FO.WRT= 000016          F.MON = 000002          ILLSW = 104565          MONPTR  003122R 025 M.SATA= 000001
FPMODE  000042R 025     F.NIL = 000001          IMUL  = ****** GX       MONSFT  003126R 025 M.SCAN= 000011
FPP   = 000000          F.NRBD= 000024          INDERR= 104410          MONTYP  003124R 025 M.SHAP= 000000
FPPLEN= 000004          F.NREC= 000030          INNTMP  003250R 025 MONVAL  003134R 025 M.SIGN= 000000
FRARGC= ****** GX       F.OPN = 000010          INNTM2  005016R 025 M.ABS = 000000          M.SUBR= 000001
FREBLK  007400R 025     F.OVBS= 000030          INPTR   000064R 025 M.BOL = 000200          M.SYM = 000203
FRESCT= ****** GX       F.PREV= 000000          INSNAM= 104462          M.BXAV= 000007      M.SYN = 000201
FUZZ    000142R 025     F.PRNT= 100000          IOERR = 104477          M.BXCR= 000007      M.TRAN= 000012
FXPTR1  003076R 025     F.PROT= 010000          I.CNT = 177776          M.BXCT= 000007      M.$ABS= 000000
F$EVAL= 000100          F.RACC= 000016          I.DIM = 177772          M.BXEX= 000007      M.$CEI= 000000
F$EVI = 000200          F.RATT= 000001          I.LEN = 000012          M.BXFX= 000007      M.$FLR= 000000
F$EXEC= 000400          F.RCNM= 000034          I.OERR= 000062          M.BXIO= 000007      M.$IDN= 000000
F$FUNC= 001000          F.RCTL= 000017          I.OLD = 177770          M.BXLC= 000003      M.$NEG= 000000
F.ACTL= 000076          F.READ= 004000          I.PTR = 177774          M.BXNC= 000007      M.$SIG= 000000
F.ALOC= 000040          F.REDF= 000040          I.STR = 177766          M.BXNL= 000007      M.1RDT= 000011
F.BBFS= 000062          F.RES = 000040          KILL  = ****** GX       M.BXPP= 000007      M.1REV= 000003
F.BDB = 000070          F.RET = 177772          LABERR= 104406          M.BXPW= 000007      M.1SCN= 000011
F.BGBC= 000057          F.RSIZ= 000002          LABLPT  005042R 025 M.BXRL= 000007          NAMBUF  003240R 025
F.BITS= 177774          F.RTYP= 000000          LARGBL  003162R 025 M.BXTS= 000007          NBRTRU= ****** GX
F.BKDN= 000026          F.R2  = 000014          LARGCT= ****** GX       M.BXWA= 000007          NB.DEV= 000200
```


APLO02 MACRO M1113 08-FEB-80 09:16 PAGE 94-3

SYMBOL TABLE

```
NB.DIR= 000100          N.PPN = 000024          RESTYP  003144R 025 SYMTP   006272R 025 T.DIM = 000100
NB.NAM= 000004          N.STAT= 000020          RESULT  003136R 025 SYMWRN= 000054        T.FUN = 010000
NB.SD1= 000400          N.UNIT= 000034          RESVAL  003154R 025 SYNERR= 104407        T.LIST= 000400
NB.SD2= 001000          OBUFLN= 001000          RETCOR= ****** GX       SYSER = ****** GX       T.LIT = 000020
NB.SNM= 000040          OCHAN   005052R 025 RINC    005104R 025 SYSERR= 104402        T.MON = 000010
NB.STP= 000020          OMODE   000402R 023 RLTBAS  005066R 025 SYSTAT  000000R 024 T.NIL = 000004
NB.SVR= 000010          OPTMP0  005066R 025 ROTINC  005070R 025 SY.BAD= 000001        T.NUM = 002000
NB.TYP= 000002          OPTMP1  005070R 025 ROUND   000010R 025 SY.BRK= 100000        T.OPFN= 000001
NB.VER= 000001          OPTMP2  005072R 025 ROVER   003312R 025 SY.BS = 000010        T.OTHR= 040000
ND.DEV= 000200          OPTMP3  005074R 025 RSTS  = 000000      SY.CLR= 000040        T.SOP = 000002
ND.EXT= 000002          OPTMP4  005076R 025 RSX   = 000115      SY.ERP= 000020        T.SPOP= 000040
ND.NAM= 000004          OPTMP5  005100R 025 RTOP    005070R 025 SY.NOU= 000100        T.TMP = 000200
ND.PPN= 000100          OPTMP6  005102R 025 RT11  = 000000      SY.OPN= 000002        T.VAL = 100000
ND.SEX= 000000          OPTMP7  005104R 025 R.BITS= 000003      SY.PMT= 001000        T.VAR = 001000
ND.SES= 000020          OPTMP8  005106R 025 R.FIX = 000001      SY.SRT= 000400        UPPER = 177400
ND.SNM= 000000          OPTMP9  005110R 025 R.LCT = 000002      SY.SVC= 000004        UTNAME  000344R 023
ND.SNS= 000040          OPTM10  005112R 025 R.LN  = 000002      SY.SVL= 000200        VALERR= 104413
ND.SPN= 001400          OPTM14  005026R 025 R.LNUM= 000010      S.AB  = 177776        VARBLE  007400R 025
NEGSGN  005063R 025 OP.MUL= ****** GX       R.NEXT= 000006      S.ASB = 100000        VARBLS  000000R 025
NINPT1  003302R 025 OP.PLU= ****** GX       R.OFF = 000000      S.BFHD= 000020        VERNUM  000000R 025
NINPT2  003300R 025 ORIGIN  000030R 025 R.PREV= 000004      S.BITS= 000002        VRSION= 000002
NINSTK  003304R 025 OUTPTR  000066R 025 R.PTR = 000000      S.CHR1= 000003        WIDTH   000072R 025
NINTMP  003310R 025 OVROPR= 000002          R.SEQ = 000003      S.FATT= 000016        WROUND  000020R 025
NINTYP  003306R 025 PATCH   007344R 025 R.VAR = 000002      S.FDB = 000140        WSDATE  000136R 025
NOTAVL= 104472          PRECSN  000052R 025 R6    =%000006      S.FLB = 040000        WSFULL= 104403
NOTOPN= 104473          RANDAC  003064R 025 R7    =%000007      S.FNAM= 000006        WSNAME  000074R 025
NPRT1   004762R 025 RANDY   003062R 025 SAME  = ****** GX       S.FNB = 000036        WSTIME  000132R 025
NULL    007364R 025 RANERR= 104411          SI    =%000005      S.FNBW= 000017        XX2     001436R 020
NUMCHN= 000020          RARGBL  003204R 025 SIDAMN= 104433      S.FNTY= 000004        XX3     001170R 020
NUMDFL  003230R 025 RARGCT= ****** GX       SIFRAM  003314R 025 S.FTYP= 000002        $CFCC = ****** GX
NUMEFL  003233R 025 RARGCV  003220R 025 SISTAK  006350R 025 S.LEN = 000002        $CLRF = ****** GX
NUMESN  003236R 025 RARGFL  003202R 025 SITOP   006346R 025 S.LINK= 177774        $DIVF = ****** GX
NUMFLG  003230R 025 RARGLE  003216R 025 SITP    007144R 025 S.NFEN= 000020        $FIS  = ****** G
NUMLEN  000062R 025 RARGLN  003210R 025 SIWARN= 000200          S.TYPE= 140000        $FPP  = ****** G
NUMNEG  003232R 025 RARGPT  003206R 025 SIZTAB  000044R 025 S.VAL = 000000        $FPPLN= ****** G
NUMQFL  003234R 025 RARGSF  003214R 025 SI.BIT= 177774          TAKCHK  001450R 020 $LDF  = ****** GX
NUMRIG  003231R 025 RARGTY  003212R 025 SI.LEN= 177772          TAKEIT  000640R 020 $MACHN= ****** G
NUMWID  003226R 025 RARGVA  003222R 025 SI.RET= 177772          TAKER   001536R 020 $MULF = ****** GX
NUMZFL  003235R 025 RBASE   005074R 025 SI.VAL= 177776          TAKER1  001546R 020 $RSTS = ****** G
NYIERR= 104432          RCOUNR  005112R 025 SKPSIZ  005134R 025 TAKER2  001664R 020 $RSX  = ****** G
N.BLSZ= 000036          RCOUNT  005110R 025 SPHOLD  000004R 024 TAKE01  001412R 020 $RT11 = ****** G
N.DESC= 000020          REDCNT  005070R 025 SPINIT  000002R 024 TKDIMS  005072R 025 $STF  = ****** GX
N.DEV = 000032          REDIGN  005066R 025 STKINC= 000774          TKFRAM  005066R 025 $SUBF = ****** GX
N.DEVN= 000034          REDOFF  005072R 025 STOP  = 104420      TKLEFT  005070R 025 $TSTF = ****** GX
N.DID = 000024          REENTR  003060R 025 SUBCNT  003104R 025 TMODE   000403R 023 .BOOLE= 000002
N.DVNM= 000032          REGTYP= 000002          SUBDCT  003106R 025 TMPTAG= 000166          .CHARA= 000000
N.EXT = 000014          RESBIT  003142R 025 SUBPTR  003110R 025 TTOUT = ****** GX       .DEF. = 000002
N.FID = 000000          RESBLK= ****** GX       SYMB1   005344R 025 TTYBUF  004140R 025 .FLOAT= 000006
N.FNAM= 000006          RESCVT  003152R 025 SYMF    005340R 025 TTYLEN= 000622          .FSRCB= ****** G
N.FTYP= 000014          RESERR= 104440          SYMPT1  005346R 025 TURKEY= 104436          .INTEG= 000004
N.FVER= 000016          RESLEN  003150R 025 SYMTAB  005352R 025 TYPE  = 000166          .MOLUN  000404R 023
N.NAME= 000006          RESPTR  003140R 025 SYMTBF= 104404          T.BOL = 020000          ...GBL= 000000
N.NEXT= 000022          RESSFT  003146R 025 SYMTOP  005350R 025 T.CON = 004000          ...TPC= 000020

. ABS.  000000  000
        000000  001
TABLES  000000  002
```

B 12

```
APLO02  MACRO M1113  08-FEB-80 09:16  PAGE 94-4
SYMBOL TABLE

MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    004224      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13875 WORDS  ( 55 PAGES)
DYNAMIC MEMORY:  15534 WORDS  ( 59 PAGES)
ELAPSED TIME:  00:06:17
APLO02,DM1:[305,301]APLO02/-SP=APLRX1,APLDEF,APLO02,APLOP,END
```

C 12

```
**FILE**ID**APLO03
```

FILEID**APL003

```
  AAAAAA      PPPPPPPP      LL              000000        000000        333333
  AAAAAA      PPPPPPPP      LL              000000        000000        333333
AA      AA    PP      PP    LL            00    00      00    00    33      33
AA      AA    PP      PP    LL            00    00      00    00    33      33
AA      AA    PP      PP    LL            00    00      00  0000          33
AA      AA    PP      PP    LL            00    00      00  0000          33
AA      AA    PPPPPPPP      LL            00    00      00  00  00      33
AA      AA    PPPPPPPP      LL            00    00      00  00  00      33
AAAAAAAAAA    PP            LL            00    00      0000    00        33
AAAAAAAAAA    PP            LL            00    00      0000    00        33
AA      AA    PP            LL            00    00      00      00    33      33      ....
AA      AA    PP            LL            00    00      00      00    33      33      ....
AA      AA    PP            LLLLLLLLLL      000000        000000        333333        ....
AA      AA    PP            LLLLLLLLLL      000000        000000        333333        ....


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                  TT
LL            SS                  TT
LL            SS                  TT
LL            SS                  TT
LL              SSSSSS            TT
LL              SSSSSS            TT
LL                    SS          TT
LL                    SS          TT
LL                    SS          TT
LL                    SS          TT
LLLLLLLLLL    SSSSSSSS            TT
LLLLLLLLLL    SSSSSSSS            TT
```


.MAIN. MACRO M1113 08-FEB-80 09:22

TABLE OF CONTENTS

.MAIN. MACRO M1113 08-FEB-80 09:22
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    50- 1060    MIXED FUNCTION           DEAL
```


APL003 MACRO M1113 08-FEB-80 09:22
TABLE OF CONTENTS

APL003 MACRO M1113 08-FEB-80 09:22
TABLE OF CONTENTS

57- 2152	MIXED FUNCTION	FIRST CO-ORDINATE REVERSE
57- 2167	MIXED FUNCTION	REVERSAL
58- 2207	MIXED FUNCTION	FIRST CO-ORDINATE ROTATE
58- 2221	MIXED FUNCTION	ROTATE
63- 2991	MIXED FUNCTION	GRADE DOWN
63- 3005	MIXED FUNCTION	GRADE UP
74- 4956	MIXED FUNCTION	MONADIC I BEAM
76- 5308	SYSTEM VARIABLE	LINE COUNTER

APL003 MACRO M1113 08-FEB-80 09:22 PAGE 1

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 1

```
                        RT11    =  0            ;NOT RT-11
                        RSTS    =  0            ;NOT RSTS/E
                        RSX     = 'M            ;RSX-11/M

                        EIS     =  0            ;NO EIS
                        FIS     =  0            ;NO FIS
                        FPP     =  0            ;NO FPP

                        FPPLEN  =  4.           ;SINGLE PRECISION

                        DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```


APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 2

G 12

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

H 12

```
APL     MACRO M1113  08-FEB-80 09:22  PAGE 42
DEFINITIONS     ERRORS
```

H 12

```
APL     MACRO M1113  08-FEB-80 09:22  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

I 12

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 44
DEFINITIONS     ERRORS
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APLOO3
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO3 IS:
      9                                 ;
     10                                 ;       APLOO3 < ...,APLOO3,APLOP
     11                                 ;
     12
     13          000003                 OVROPR= 3                       ;OVERLAY SPECIFIED OPERATORS
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 46
DEFINITIONS     ERRORS
```

```
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                        ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                        ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                        ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                        ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                        ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                        ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                        ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                        ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                        ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                        ;       INTACT!
                        ;--


                        ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                        ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                        ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                        ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                        ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                        ;MACROS.

                        .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                        .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                        .IF EQ OVROPR-D.'NAME
                                .PSECT  CODE
                        D$'NAME::
                        .ENDM

                        .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                        .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                        .IF EQ OVROPR-M.'NAME
                                .PSECT  CODE
                                .IF NE NUM
                        M$'NAME::
                            .ENDC
                        .ENDM

                        .MACRO  LIST
                                .ENDC
                        .ENDM

                        .MACRO  PRMRES
                                .IF EQ 1
                        .ENDM

                                        ;************ ASSEMBLING THE FILE!!!!!
007400                  PRMRES
                                LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,.INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,.INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,.INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,.FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 47-1
DEFINITIONS ERRORS

```
    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```


APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 48
DEFINITIONS ERRORS

```
    641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
    652 007400                                  LIST
    653
    654
    655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
    662 007400                                  LIST
    663
    664
    665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
    676 007400                                  LIST
    677
    678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
    687 007400                                  LIST
    688
    689
    690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
    700 007400                                  LIST
    701
    702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    712 007400                                  LIST
    713
    714
    715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    733 007400                                  LIST
    734
    735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    745 007400                                  LIST
    746
    747
    748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
    758 007400                                  LIST
    759
    760
    761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
    780 007400                                  LIST
    781
```

DEFINITIONS ERRORS

```
783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
786 007400                                  LIST
787
788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
791 007400                                  LIST
792
793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
799 007400                                  LIST
800
801
802 007400                          MOPINS  $IDN,-2
808 007400                                  LIST
809
810
811 007400                          MOPINS  $FLR,-14
855 007400                                  LIST
856
857
858 007400                          MOPINS  $CEI,-16
868 007400                                  LIST
869
870
871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
878 007400                                  LIST
879
880 007400                          MOPINS  $NEG,-4
887 007400                                  LIST
889
889
890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
901 007400                                  LIST
902
903 007400                          MOPINS  $SIG,-6
914 007400                                  LIST
915
916
917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
928 007400                                  LIST
929
930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
939 007400                                  LIST
940
941
942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
949 007400                                  LIST
950
951
952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
959 007400                                  LIST
960
961 007400                          PRMRES
966 007400                                  LIST
967
968
969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
977 007400                                  LIST
978
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,.FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 50
DEFINITIONS     ERRORS
```

```
1059 007400                                DOPINS  DEAL,100
1060                                       .SBTTL  MIXED FUNCTION          DEAL
1061
1062 000000  013702  003162'               MOV     LARG3L,R2       ;Pointer to Left Arg
1063 000004  022762  000001  000004        CMP     #1,A.ALEN(R2)   ;Is it a Scalar?
1064 000012  001052                        BNE     1$              ;Nope, RANK ERROR
1065 000014  013700  003170'               MOV     LARGTY,R0       ;Get Left Arg Type
1066 000020  001450                        BEQ     2$              ;Wrong, DOMAIN ERROR
1067 000022  011202                        MOV     (R2),R2         ;Point to Self
1068 000024                                CALL    @ILARGC(R0)     ;Fetch Length (integer)
1069 000030  013700  003200'               MOV     LARGVA,R0       ;Get The Length
1070 000034  002442                        BLT     2$              ;Illegal, DOMAIN ERROR
1071 000036  013703  003204'               MOV     RARGBL,R3       ;Right Arg Data
1072 000042  005037  003202'               CLR     RARGFL          ;Clear the increment flag
1073 000046  022763  000001  000004        CMP     #1,A.ALEN(R3)   ;Must be a Scalar
1074 000054  001031                        BNE     1$              ;Nope, RANK ERROR
1075 000056  013701  003212'               MOV     RARGTY,R1       ;Get Right Arg Type
1076 000062  001427                        BEQ     2$              ;Wrong, DOMAIN ERROR
1077 000064  011303                        MOV     (R3),R3         ;Point to data
1078 000066                                CALL    @FRARGC(R1)     ;Fetch Right Arg
1079 000072  020127  000004                CMP     R1,#.INTEGER    ;Verify it is Integer
1080 000076  001553                        BEQ     7$              ;It is, check for Special
1081 000100  002421                        BLT     3$              ;No, But is Boolean
1082 000102                                STF     AC2,AC0         ;Copy Value
1083 000110                                CFCC                    ;Is it Positive?
1084 000114  003412                        BLE     2$              ;No, DOMAIN ERROR
1085 000116                                CALL    F.TO.I          ;Convert (if possible)
1086 000122  000410                        BR      3$              ;Not so, Cannot use Special Case
1087
1088 000124  013737  003154' 003222'       MOV     RESVAL,RARGVA   ;It did, use
1089 000132  012701  000004                MOV     #.INTEGER,R1    ;  Integer Value
1090 000136  000535                        BR      8$              ;Try Special Case
1091
1092 000140  104411                1$:     RANERR                  ;RANK ERROR
1093 000142  104417                2$:     DOMERR                  ;DOMAIN ERROR
1094 000144  010137  003144'       3$:     MOV     R1,RESTYP       ;Result Type
1095 000150  013700  003200'               MOV     LARGVA,R0       ;Get Length
1096 000154                                LDCIF   R0,AC1          ;Get Floating Form
1097 000162                                CMPF    AC1,AC2         ;Which is longer
1098 000170                                CFCC
1099 000174  003362                        BGT     2$              ;DOMAIN ERROR
1100 000176  010002                        MOV     R0,R2           ;SAVE LENGTH
1101 000200                                ASH     SIZTAB(R1),R0   ;CONVERT TO BYTES
1102 000214  062700  000010                ADD     #A.DIM1+2,R0    ;ADD IN OVERHEAD
1103 000220                                CALL    GETFRA          ;Get result, SI frame for it
1104 000224  062720  000010                ADD     #A.DIM1+2,(R0)+ ;ADD IN OFFSET
1105 000230  112720  000001                MOVB    #1,(R0)+        ;1 DIMENSION
1106 000234  110110                        MOVB    R1,(R0)         ;SET THE TYPE
1107 000236  152720  000200                BISB    #A.ASB,(R0)+    ;  AND THE FLAG
1108 000242  010220                        MOV     R2,(R0)+        ;SET THE ALEN
1109 000244  010220                        MOV     R2,(R0)+        ;AND THE 1ST DIM
1110 000246  001002                        BNE     13$             ;The real target is too far away
1111 000250  000137  000654'               JMP     12$             ;Nothing there, Finished
1112 000254  016137  000000G 003152' 13$:  MOV     FRESCT(R1),RESCVT ;RESULT STORE ROUTINE
1113 000262                                STF     AC2,AC3         ;COPY FP ARG OVER
1114 000270  016137  000000G 005122'       MOV     ECOMCT(R1),COMPTR ;COMPARE ROUTINE POINTER
1115 000276  016137  000054' 003160'       MOV     LENTAB(R1),LARGFL ;LEFT ARG INCREMENT
```

```
1116 000304                                    MUL     #5.,R1              ;LOOK ALONG DIAGONAL
1117 000320  016137  000000G 003176'          MOV     LARGCT(R1),LARGCV ;LEFT FETCH
1118 000326  016137  000000G 003220'          MOV     RARGCT(R1),RARGCV ;RIGHT FETCH
1119 000334  010037  003140'                   MOV     R0,RESPTR           ;SAVE RESULT POINTER
1120 000340  010001                            MOV     R0,R1               ;RESULT POINTER IN R1
1121 000342  010204                            MOV     R2,R4               ;GET LOOP COUNT
1122 000344                            4$:     STF     AC3,AC0             ;GET ARG
1123 000352                                    CALL    .RAND               ;GET RANDOM VALUE
1124 000356  010103                            MOV     R1,R3               ;SAVE CURRENT RESULT POINTER
1125 000360                                    CALL    @RESCVT             ;STORE RESULT
1126 000364  104402                            SYSERR                      ;THE SHIP NA CANNA TAKE MUCH MORE, CAPTAIN
1127 000366  013702  003140'                   MOV     RESPTR,R2           ;POINT TO BEGINNING
1128 000372                                    CALL    @RARGCV             ;FETCH CURRENT RESULT VALUE
1129 000376  020203                    5$:     CMP     R2,R3               ;STILL IN RANGE?
1130 000400  001407                            BEQ     6$                  ;NO! NOT THE WHIP! ANYTHING BUT THE WHIP!
1131 000402                                    CALL    @LARGCV             ;FETCH PREVIOUS RESULTS
1132 000406                                    CALL    @COMPTR             ;COMPARE 'EM
1133 000412  001371                            BNE     5$                  ;LOOP
1134 000414  010301                            MOV     R3,R1               ;RESTORE RESULT POINTER
1135 000416  000752                            BR      4$                  ;AND TRY AGAIN
1136 000420                            6$:     SOB     R4,4$               ;LOOP FOR ONE AND ALL
1137 000424                                    RETURN                      ;FINISHED
1138
1139 000426                            7$:     CALL    @IRARGC(R1)         ;FETCH RIGHT ARG
1140 000432  013700  003222'           8$:     MOV     RARGVA,R0           ;GET THE LARGEST VAL
1141 000436  013702  003200'                   MOV     LARGVA,R2           ;GET THE LENGTH
1142 000442  020200                            CMP     R2,R0               ;RIGHT SHOULD BE BIGGER
1143 000444  003236                            BGT     2$                  ;WRONG! GIVE UP NOW, AVOID THE RUSH
1144 000446  010003                            MOV     R0,R3               ;COPY LARGEST VAL
1145 000450  006203                            ASR     R3                  ;DIVIDE BY 2
1146 000452  006203                            ASR     R3                  ;   ''    '' 4
1147 000454  006203                            ASR     R3                  ;   ''    '' 8
1148 000456  020203                            CMP     R2,R3               ;IS IT BIGGER?
1149 000460  002631                            BLT     3$                  ;NO, USE NORMAL METHODS
1150 000462  010003                            MOV     R0,R3               ;SAVE COUNT
1151 000464  006300                            ASL     R0                  ;(WORDS)
1152 000466  062700  000010                   ADD     #A.DIM1+2,R0        ;  PLUS OVERHEAD
1153 000472                                    CALL    GETCO1              ;ATTEMPT CALL
1154 000476  000622                            BR      3$                  ;FAILED, TRY W/ NORMAL ROUTINE
1155 000500                                    SIFRAM                      ;Build an SI frame
1156 000512  010010                            MOV     R0,(R0)             ;POINT TO SELF
1157 000514  062720  000010                    ADD     #A.DIM1+2,(R0)+     ;POINT TO DATA
1158 000520  012720  102001                    MOV     #<A.ASB+.INTEGER>*400+1,(R0)+ ;BITS/1 DIM
1159 000524  010220                            MOV     R2,(R0)+            ;ALEN
1160 000526  010220                            MOV     R2,(R0)+            ;1ST DIM
1161 000530  001447                            BEQ     11$                 ;NOTHING, FINISHED
1162 000532  010004                            MOV     R0,R4               ;COPY RESULT POINTER
1163 000534                                    PUSH    R3                  ;SAVE WORK COUNT
1164 000536  013701  000030'                   MOV     ORIGIN,R1           ;GET ORIGIN (0 OR 1)
1165 000542  010120                    9$:     MOV     R1,(R0)+            ;PLACE A NUMBER
1166 000544  005201                            INC     R1                  ;INCREMENT COUNT
1167 000546                                    SOB     R3,9$               ;LOOP FOR WORK DATA
1168 000552  011602                            MOV     (SP),R2             ;GET THE WORK COUNT
1169 000554  010403                            MOV     R4,R3               ;POINT TO RESULT/SOURCE
1170 000556  005001                    10$:    CLR     R1                  ;DIVIDE IS FUNNY
1171 000560  013700  003062'                   MOV     RANDY,R0            ;RANDOM VALUE
1172 000564  042700  100000                   BIC     #100000,R0          ;CLEAR HIGH BIT
```

```
1173 000570                                    MUL     @SP,R0          ;DOUBLE PRECISION MULTIPLY
1174 000602                                    ASHC    #1,R0           ;GET INTEGER RANDOM NUMBER
1175 000616  006300                            ASL     R0              ;WORD OFFSET
1176 000620  060400                            ADD     R4,R0           ;POINT TO RANDOM ELEMENT
1177 000622  011301                            MOV     (R3),R1         ;GET THE CURRENT ELEMENT
1178 000624  011023                            MOV     (R0),(R3)+      ;COPY OTHER VALUE
1179 000626  010110                            MOV     R1,(R0)         ;FINISH EXCHANGE
1180 000630                                    LDCIF   R1,AC0          ;GET VALUE
1181 000636                                    CALL    .RAND           ;RANDOM NUMBER
1182 000642                                    SOB     R2,10$          ;LOOP FOR EACH ELEMENT
1183 000646  005726                            TST     (SP)+           ;SKIP GARBAGE
1184 000650                            11$:    CALL    CARVE           ;CLEAN IT UP
1185 000654                            12$:    RETURN                  ;FINISHED
1187 000656                                    LIST
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 51
MIXED FUNCTION          DEAL

 1189 000656                          MOPINS  SHAP,102
 1211 000656                                  LIST
 1212
 1213
 1214
 1215
 1216
 1217 000656                          DOPINS  RHO,102
 1313 000656                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 52
MIXED FUNCTION          DEAL
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 52
MIXED FUNCTION          DEAL

   1315 000656                          MOPINS  RAVL,56
   1325 000656                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 000656                          DOPINS  CTLM,56,T.DIM
   1596 000656                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 53
MIXED FUNCTION          DEAL
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 53
MIXED FUNCTION          DEAL

   1598 000656                          MOPINS  IOTA,72
   1632 000656                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 000656                          DOPINS  IND,72
   1699 000656                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 54
MIXED FUNCTION          DEAL
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 54
MIXED FUNCTION          DEAL

   1701 000656                          DOPINS  1COM,124
   1709 000656                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 000656                          DOPINS  COMP,60,T.DIM
   1789 000656                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 55
MIXED FUNCTION          DEAL
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 55
MIXED FUNCTION          DEAL

   1791 000656                          DOPINS  1EXP,130
   1799 000656                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 000656                          DOPINS  EXPD,62,T.DIM
   1864 000656                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 56
MIXED FUNCTION          DEAL
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 56
MIXED FUNCTION          DEAL

   1866 000656                          DOPINS  DROP,104
   1907 000656                                  LIST
   1908
   1909
   1910
   1911 000656                          DOPINS  TAKE,114
   2149 000656                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 57
MIXED FUNCTION          DEAL
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 57
MIXED FUNCTION          DEAL

                                        MOPINS  1REV,146
                                        .SBTTL  MIXED FUNCTION          FIRST CO-ORDINATE REVERSE

000656  013704  003120'                        MOV     MONBLK,R4       ;POINT TO ARG
000662  116401  000003                        MOVB    A.BITS(R4),R1   ;GET ARRAY'S BITS
000666  012702  000001                        MOV     #1,R2           ;USE FIRST CO-ORDINATE
000672                                        CALL    FIXDIM          ;GET DIM POINTER
000676  000137  000706'                        JMP     REVRS1          ;DO REVERSAL
000702                                        LIST


                                        MOPINS  REVS,144,T.DIM
                                        .SBTTL  MIXED FUNCTION          REVERSAL

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       REVERSE IS VERY SIMPLE! AS SIMPLE AS 'ABC'.  JUST REMEMBER THERE
                                        ;       ARE 'A' GROUPS OF 'B' TO BE REVERSED.  THE GROUPS OF 'B' ARE REVERSED
                                        ;       IN A BUT THE 'C' ELEMENTS IN EVERY 'B' KEEP THE SAME ORDER.  'A' IS
                                        ;       THE PRODUCTS OF ALL DIMS PRECEDING THE REVERSAL DIM AND 'C' IS THE
                                        ;       PRODUCT OF ALL DIMS FOLLOWING THE REVERSAL DIM.  'B' IS THE SIZE OF
                                        ;       THE REVERSAL DIM.
                                        ;--

000702                                  REVRS1: CALL    MFXDIM          ;GET DIM TO USE
000706  017701  005122'                        MOV     @COMPTR,R1      ;KEEP SAME SIZE ALONG DIM
000712                                        CALL    RESBLK          ;CREATE RESULT BLOCK
000716  017703  005122'                        MOV     @COMPTR,R3      ;GET 'B'
000722  005764  000004                        TST     A.ALEN(R4)      ;Anything there?
000726  001431                                BEQ     5$              ;DONE IF NO LEN
000730  013702  003122'                        MOV     MONPTR,R2       ;POINT TO START OF SOURCE DATA
000734  063702  005130'                        ADD     CNTSIZ,R2       ;POINT TO SECOND 'A' GROUP
000740  163702  005134'                        SUB     SKPSIZ,R2       ;BUT MOVE DOWN TO LAST 'B' IN FIRST 'A'

                                        ;
                                        ; DO REVERSAL WITH SET UP PARAMETERS
                                        ;

000744  010201                          2$:     MOV     R2,R1           ;POINT TO LAST 'B' IN 'A' OF SOURCE
000746  017703  005122'                        MOV     @COMPTR,R3      ;GET # OF 'B'S
000752  013704  005134'                  3$:   MOV     SKPSIZ,R4       ;AND SIZE OF 'B' IN BYTES
000756  112120                          4$:   MOVB    (R1)+,(R0)+     ;COPY DATA
000760                                        SOB     R4,4$           ;COPY ALL OF THE 'B'
000764  163701  005134'                        SUB     SKPSIZ,R1       ;BACK UP 1 'B' IN SOURCE
000770  163701  005134'                        SUB     SKPSIZ,R1       ;THEN ONE MORE 'B'
000774                                        SOB     R3,3$           ;DO ALL 'B'S IN THIS 'A'
001000  063702  005130'                        ADD     CNTSIZ,R2       ;MOV SOURCE POINTER TO NEXT A
001004  005337  005126'                        DEC     COMCNT          ;MORE 'A'S
001010  001355                                BNE     2$              ;YES
001012                                  5$:   RETURN                  ;GRAND FINALLE
001014                                        LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 58
MIXED FUNCTION          REVERSAL
```

```
2206 001014                                DOPINS  1ROT,146
2207                                       .SBTTL  MIXED FUNCTION          FIRST CO-ORDINATE ROTATE
2208
2209 001014                                        CALL    GETREG          ;LOAD SOME REGISTERS
2210 001020  012702  000001                   MOV     #1,R2           ;USE DIM 1
2211 001024                                    CALL    FIXDIM          ;SET UP DIM POINTERS
2212 001030  000137  001040'                   JMP     ROTAT1          ;ROTATE IT!
2214 001034                                    LIST
2215
2216
2217
2218
2219
2220 001034                                DOPINS  ROT,144,T.DIM
2221                                       .SBTTL  MIXED FUNCTION          ROTATE
2222
2223                                       ;ROTATE, LIKE REVERSAL, IS AS EASY AS 'ABC'
2224
2225 001034                                        CALL    DFXDIM          ;GET THE DIM
2226 001040  017701  005122'          ROTAT1: MOV     @COMPTR,R1      ;NO CHANGE IN SIZE
2227 001044                                    CALL    RESBLK          ;CREATE RESULT BLOCK
2228 001050  010037  003136'                   MOV     R0,RESULT       ;SAVE POINTER
2229 001054  013700  003162'                   MOV     LARGBL,R0       ;POINT TO LEFT ARG
2230 001060  013702  003170'                   MOV     LARGTY,R2       ;GET THE LEFT ARG TYPE
2231 001064  001001                            BNE     5$              ;YES, IT'S NUMERIC
2232 001066  104417                            DOMERR                  ;NO, DOMAIN ERROR
2233 001070  016237  000000G 003220' 5$:      MOV     IRARGC(R2),RARGCV ;Used to fetch the Left argument
2234 001076  005037  003202'                   CLR     RARGFL          ;ASSUME SCALAR
2235 001102  105760  000002                   TSTB    A.DIMS(R0)      ;LEFT A SCALAR?
2236 001106  001444                            BEQ     4$              ;YES
2237
2238 001110  116002  000002                   MOVB    A.DIMS(R0),R2   ;GET # OF DIMS IN LEFT
2239 001114  013701  003204'                   MOV     RARGBL,R1       ;POINT TO RIGHT ARG
2240 001120  116103  000002                   MOVB    A.DIMS(R1),R3   ;GET # OF RIGHT DIMS
2241 001124  005303                            DEC     R3              ;DON'T COUNT ONE
2242 001126  002403                            BLT     1$              ;ERROR! TOO FEW DIMS!
2243 001130  001403                            BEQ     13$             ;Right argument is a vector, check special case
2244 001132  020203                            CMP     R2,R3           ;SAME NUMBER OF DIMS (ADJUSTED)
2245 001134  001411                            BEQ     2$              ;YES
2246 001136  104411                  1$:       RANERR                  ;RANK ERROR
2247 001140  020227  000001          13$:      CMP     R2,#1           ;Is Left argument a vector?
2248 001144  001374                            BNE     1$              ;No
2249 001146  026027  000004  000001           CMP     A.ALEN(R0),#1   ;Is Left argument a singleton vector?
2250 001154  001370                            BNE     1$              ;No
2251 001156  000420                            BR      4$              ;Yes, Then skip length check
2252 001160  062700  000006          2$:       ADD     #A.DIM1,R0      ;POINT TO LEFT DIMS
2253 001164  062701  000006                   ADD     #A.DIM1,R1      ;POINT TO RIGHT DIMS
2254 001170  020137  005122'          3$:       CMP     R1,COMPTR       ;IS THIS THE DIM TO ROTATE ALONG?
2255 001174  001001                            BNE     9$              ;NO
2256 001176  005721                            TST     (R1)+           ;SKIP TO NEXT DIM OF RIGHT
2257 001200  022021                  9$:       CMP     (R0)+,(R1)+     ;SAME SIZE DIMS?
2258 001202  001401                            BEQ     6$              ;YES
2259 001204  104412                            LENERR                  ;LENGTH ERROR
2260 001206                          6$:       SOB     R2,3$           ;COMPARE REST OR DIMS
2261 001212  013737  003160' 003202'           MOV     LARGFL,RARGFL   ;LEFT ARG INC
2262
2263 001220  005777  005122'          4$:       TST     @COMPTR         ;ANY ELEMENTS ALONG DIM?
```

```
2264 001224  003472                                    BLE     12$             ;YES, ALREADY DONE
2265 001226  017737  005122' 003200'                    MOV     @COMPTR,LARGVA  ;FLOAT DIM SIZE ('B')
2266                                                    ;
2267                                                    ;DO ROTATION ALREADY! QUIT STALLING AND SETTING UP! COM'ON!
2268                                                    ;
2269 001234  013737  003136' 005066'  7$:      MOV     RESULT,RLTBAS   ;SET UP RESULT BASE
2270 001242  013700  003206'                            MOV     RARGPT,R0       ;POINT TO RIGHT ARG
2271 001246  013737  005124' 005072'                    MOV     COMSKP,CCNT     ;SET UP FOR COUNT OF 'C'
2272                                                    ;
2273                                                    ;FIX UP FOR 'B' ELEMENTS 'C' APART
2274                                                    ;
2275 001254  013704  005066'          8$:      MOV     RLTBAS,R4       ;POINTER TO CURRENT RESULT BASE
2276 001260  013703  003164'                            MOV     LARGPT,R3       ;POINT TO LEFT ARG
2277 001264                                             CALL    @RARGCV         ;ARG FETCH
2278 001270  010337  003164'                            MOV     R3,LARGPT       ;SAVE POINTER TO LEFT ARG
2279 001274                                             CALL    D$RES           ;GET THE RESIDUE (MOD THE CRITTER)
2280 001300  013701  003154'                            MOV     RESVAL,R1       ;AND FIX IT
2281 001304  017702  005122'                            MOV     @COMPTR,R2      ;GET SIZE ALONG DIM
2282 001310  160102                                     SUB     R1,R2           ;MAKE IT '# TO LEFT'
2283 001312  010103                                     MOV     R1,R3           ;GET '# TO RIGHT'
2284 001314                                             MUL     SKPSIZ,R3       ;MAKE OFFSET TO '# TO LEFT'
2285 001330  060003                                     ADD     R0,R3           ;MAKE ABSOLUTE
2286 001332                                             CALL    MOVROT          ;ROTATAE THE DATA
2287
2288 001336  010102                                     MOV     R1,R2           ;COPY '# TO RIGHT'
2289 001340  001403                                     BEQ     10$             ;IF NONE TO MOVE, DON'T MOVE NONE
2290 001342  010003                                     MOV     R0,R3           ;COPY SOURCE BASE
2291 001344                                             CALL    MOVROT          ;ROTATE THE DATA
2292
2293 001350  063700  005132'          10$:     ADD     ELESIZ,R0       ;BUMP SOURCE BASE TO NEXT ELEMENT
2294 001354  063737  005132' 005066'                    ADD     ELESIZ,RLTBAS   ;BUMP RESULT BASE TO NEXT ELEMENT
2295 001362  005337  005072'          11$:     DEC     CCNT            ;ANYMORE WITHIN THE 'B'
2296 001366  001332                                     BNE     8$              ;YES
2297                                                    ;
2298                                                    ;JUMP TO NEXT 'A'
2299                                                    ;
2300 001370  063737  005130' 003136'                    ADD     CNTSIZ,RESULT   ;NEXT RESULT 'A'
2301 001376  063737  005130' 003206'                    ADD     CNTSIZ,RARGPT   ;NEXT SOURCE 'A'
2302 001404  005337  005126'                            DEC     COMCNT          ;ANY MORE 'A'S?
2303 001410  001311                                     BNE     7$              ;YES
2304 001412                                    12$:     RETURN                  ;ALL DONE
2305
2306                                                    ;'MOVROT' IS USED TO MOVE THE DATA.  IT MOVES R2 ELEMENTS STARTING AT
2307                                                    ;R3 TO R4, INCREMENTING BY 'SKPSIZ'
2308
2309 001414                                    MOVROT: PUSH    R0              ;SAVE R0
2310
2311 001416                                    1$:      PUSH    <R3,R4>         ;SAVE REGISTERS
2312 001422  013700  005132'                            MOV     ELESIZ,R0       ;GET SIZE OF ELEMENT
2313
2314 001426  112324                            2$:      MOVB    (R3)+,(R4)+     ;MOVE CHAR
2315 001430                                             SOB     R0,2$           ;LOOP FOR ELEMNT
2316 001434                                             POP     <R4,R3>         ;RESTORE THE REGISTERS
2317 001440  063703  005134'                            ADD     SKPSIZ,R3       ;NEXT SOURCE ELEMENT
2318 001444  063704  005134'                            ADD     SKPSIZ,R4       ;NEXT RESULT ELEMENT
2319 001450                                             SOB     R2,1$           ;DO ALL OF 'EM WANTED
2320
```

```
2321 001454                             POP     R0              ;RESTORE R0
2322 001456                             RETURN                  ;ALL DONE
2323
2324                            ;RLTBAS= OPTMP0                 ;RESULT BASE
2325                            ;ROTINC= OPTMP1                 ;ROTATION'S INC FOR INDEX OF LEFT ARG
2326                            ;CCNT=   OPTMP2                 ;COUNT OF 'C' GROUPS
2328 001460                             LIST
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 59
MIXED FUNCTION          ROTATE

   2330 001460                          MOPINS  TRAN,126
   2415 001460                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 60
MIXED FUNCTION          ROTATE
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 60
MIXED FUNCTION          ROTATE

  2417 001460                          DOPINS  TRAN,126
  2654 001460                                  LIST
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 61
MIXED FUNCTION          ROTATE
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 61
MIXED FUNCTION          ROTATE

   2656 001460                          DOPINS  ENCD,76
   2856 001460                                  LIST
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 62
MIXED FUNCTION          ROTATE
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 62
MIXED FUNCTION          ROTATE

   2858 001460                          DOPINS  DECD,64
   2988 001460                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 63
MIXED FUNCTION          ROTATE
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 63
MIXED FUNCTION          ROTATE

 2990 001460                            MOPINS  GDWN,136,T.DIM
 2991                                   .SBTTL  MIXED FUNCTION          GRADE DOWN
 2992
 2993                                   ;+
 2994                                   ;CALLS SORT ROUTINE TO RETURN INDICES OF THE ARGUMENT SORTED
 2995                                   ;  INTO DECENDING ORDER.
 2996                                   ;-
 2997
 2998 001460  012737  177776  005102'          MOV     #-2,OPTMP6      ;SET THE GRADE DOWN FLAG
 2999 001466  000137  001500'                  JMP     SORT            ;AND CALL THE SORTER
 3001 001472                                   LIST
 3002
 3003
 3004 001472                            MOPINS  GDUP,140,T.DIM
 3005                                   .SBTTL  MIXED FUNCTION          GRADE UP
 3006
 3007                                   ;++
 3008                                   ; GRADE UP/DOWN ARE IMPLEMENTED USING A LINKED TWO-WAY MERGE SORT WHICH
 3009                                   ; PRODUCES A SORTED CHAIN OF INDICES.  THIS CHAIN IS THEN TRANSFORMED INTO
 3010                                   ; A PERMUTATION VECTOR, WHICH IS THEN INVERTED TO OBTAIN THE DESIRED RESULT.
 3011                                   ; THE ALGORITHM RUNS IN O(LOG N) TIME AND REQUIRES 2*LOG N TEMPORARY SPACE.
 3012                                   ; THIS TEMPORARY SPACE IS ALLOCATED ON THE SP STACK.  THE MAXIMUM NEEDED,
 3013                                   ; 100 OCTAL BYTES, IS ALWAYS RESERVED.  THE ALGORITHM IS DUE TO L. J. WOODRUM,
 3014                                   ; IBM SYSTEM JOURNAL, 8(1969) 189-203.  SEE ALSO D. K. KNUTH, THE ART OF
 3015                                   ; COMPUTER PROGRAMMING VOL 3, SORTING AND SEARCHING, 159-170, AND R. HART,
 3016                                   ; CREATIVE COMPUTING, JAN/FEB 1978, 96-101 FOR POSSIBLE IMPROVEMENTS.
 3017                                   ;
 3018                                   ; REWRITTEN:    MAR 27, 1978    BY: PETER HAYNES
 3019                                   ;
 3020                                   ; REGISTER ASSIGNMENTS:
 3021                                   ;
 3022                                   ;       OPTMP1  =       DIMENSION TO SORT ALONG, DEFAULT = 1
 3023                                   ;       OPTMP2  =       STRIDE = OPTMP3*OPTMP4
 3024                                   ;       OPTMP3  =       SECOND DIMENSION, IF VECTOR THEN EQUALS 1
 3025                                   ;       OPTMP4  =       SIZE IN BYTES OF ONE ELEMENT
 3026                                   ;       OPTMP5  =       THE NEGATION OF THE ORIGIN
 3027                                   ;       OPTMP6  =       GRADE UP/DOWN FLAG, USED AS AN OFFSET BY THE GDCMP RTN
 3028                                   ;                         GRADE UP: 2,  GRADE DOWN: -2
 3029                                   ;       OPTMP7  =       LENGTHS STACK POINTER
 3030                                   ;       OPTMP8  =       HEADERS STACK POINTER
 3031                                   ;       OPTMP9  =       NEX, THE NEXT ELEMENT OF THE PERM VECTOR TO MERGE
 3032                                   ;--
 3033
 3034 001472  012737  000002  005102'          MOV     #2,OPTMP6       ;SET THE GRADE UP FLAG
 3035 001500  005037  005070'          SORT:   CLR     OPTMP1          ;ASSUME NULL ARGUMENT
 3036 001504  013703  003120'                  MOV     MONBLK,R3       ;POINT TO ARGUMENT BLOCK
 3037 001510  005763  000004                   TST     A.ALEN(R3)      ;NULL ARGUMENT?
 3038 001514  001473                           BEQ     3$              ;YES, RETURN THE NULL VECTOR
 3039 001516  116304  000002                   MOVB    A.DIMS(R3),R4   ;GET RANK OF THE ARGUMENT
 3040 001522                                   CONINC  R4              ;CONVERT SCALAR TO VECTOR
 3041 001526  012737  000001  005074'          MOV     #1,OPTMP3       ;DEFAULT FOR 2ND DIMENSION OF VECTOR
 3042 001534  020427  000002                   CMP     R4,#2           ;TOO MANY DIMENSIONS?
 3043 001540  003401                           BLE     1$              ;NO, OK
 3044 001542  104411                           RANERR                  ;YES, RANK ERROR
 3045 001544  002403                   1$:     BLT     2$              ;VECTOR, NO SECOND DIEMENSION
 3046 001546  016337  000010  005074'          MOV     A.DIM1+2(R3),OPTMP3     ;SECOND DIMENSION
 3047 001554  016337  000006  005070'  2$:     MOV     A.DIM1(R3),OPTMP1       ;FIRST DIMENSION
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 63-1
MIXED FUNCTION          GRADE UP
```

```
3048 001562 013702 003124'                      MOV     MONTYP,R2               ;GET ARGUMENT TYPE
3049 001566 016201 000054'                      MOV     LENTAB(R2),R1           ;OFFSET FOR SECOND DIMENSION
3050 001572 016237 000000G 003156'              MOV     ECOMCT(R2),COMRTN       ;POINTER TO COMPARE ROUTINE
3051 001600 006302                              ASL     R2                      ;TIMES 2
3052 001602 006302                              ASL     R2                      ;TIMES 4
3053 001604 063702 003124'                      ADD     MONTYP,R2               ;TIMES 5
3054 001610 016237 000000G 003176'              MOV     LARGCT(R2),LARGCV       ;POINTER TO LEFT FETCH ROUTINE
3055 001616 016237 000000G 003220'              MOV     RARGCT(R2),RARGCV       ;POINTER TO RIGHT FETCH ROUTINE
3056 001624 010137 005076'                      MOV     R1,OPTMP4       ;ELEMENT SIZE, STRIDE FOR FIRST DIMENSION
3057 001630                                     MUL     OPTMP3,R1       ;CALCULATE STRIDE
3058 001644 010137 005072'                      MOV     R1,OPTMP2       ;SAVE IT
3059 001650                                     CALL    MONDIM          ;DIMENSIONAL OPERATOR
3060 001654 020204                              CMP     R2,R4           ;SAME AS LAST DIMENSION?
3061 001656 001412                              BEQ     3$              ;YES, NO PROBLEM
3062 001660 012700 005072'                      MOV     #OPTMP2,R0      ;POINT TO FIRST DIMENSION DATA
3063 001664 012701 005076'                      MOV     #OPTMP4,R1      ;POINT TO SECOND DIMENSION DATA
3064 001670 011002                              MOV     (R0),R2         ;EXCHANGE INCREMENTS
3065 001672 011110                              MOV     (R1),(R0)
3066 001674 010211                              MOV     R2,(R1)
3067 001676 014002                              MOV     -(R0),R2        ;EXCHANGE DIMENSION SIZES
3068 001700 014110                              MOV     -(R1),(R0)
3069 001702 010211                              MOV     R2,(R1)
3070 001704 013700 005070'              3$:     MOV     OPTMP1,R0       ;GET SORT DIMENSION
3071 001710 006300                              ASL     R0              ;COMPUTE RESULT SIZE IN WORDS
3072 001712 062700 000010                       ADD     #A.DIM1+2,R0    ;ADD IN OVERHEAD
3073 001716                                     CALL    GETFRA          ;Get result, SI frame for it
3074 001722 010037 000000G                      MOV     R0,RESBLK       ;SAVE POINTER TO RESULT
3075 001726 062720 000010                       ADD     #10,(R0)+               ;SET POINTER TO DATA
3076 001732 112720 000001                       MOVB    #1,(R0)+                ;RESULT IS VECTOR
3077 001736 112720 000204                       MOVB    #A.ASB+.INTEGER,(R0)+   ;VALUE IS NUMERIC
3078 001742 013702 005070'                      MOV     OPTMP1,R2               ;SORT DIMENSION (LOOP COUNT)
3079 001746 010220                              MOV     R2,(R0)+                ;ALEN
3080 001750 010220                              MOV     R2,(R0)+                ;DIMENSION 1
3081 001752 010037 003140'                      MOV     R0,RESPTR       ;SAVE POINTER TO RESULT DATA
3082 001756 026327 000004  000001               CMP     A.ALEN(R3),#1   ;TEST FOR TRIVAL CASES
3083 001764 003004                              BGT     5$              ;NON-TRIVAL
3084 001766 002402                              BLT     4$              ;RESULT IS NULL VECTOR, QUIT NOW
3085 001770 013710 000030'                      MOV     ORIGIN,(R0)     ;RESULT IS ORIGIN
3086 001774                             4$:     RETURN                  ;AND RETURN
3087
3088 001776 012737 000001  005100' 5$:  MOV     #1,OPTMP5       ;COMPLEMENT THE ORIGIN
3089 002004 163737 000030' 005100'      SUB     ORIGIN,OPTMP5   ;NOTORG = 1-ORIGIN
3090 002012 013737 005076' 003160'      MOV     OPTMP4,LARGFL   ;GET INCREMENTS FOR COMPARSION ROUTINE
3091 002020 013737 005076' 003202'      MOV     OPTMP4,RARGFL
3092 002026 005746                      TST     -(SP)           ;RESERVE SPACE FOR THE TEMP STACKS
3093 002030 010600                      MOV     SP,R0           ;       LPTR=1
3094 002032 010037 005104'              MOV     R0,OPTMP7       ;SAVE LPTR FOR LATER COMPARISON
3095 002036 162706 000100               SUB     #100,SP         ;SKIP OVER THE STACK AREA
3096 002042 010637 005106'              MOV     SP,OPTMP8       ;       HPTR=0
3097 002046 005726                      TST     (SP)+           ;GET SP RIGHT
3098 002050 013710 005070'              MOV     OPTMP1,(R0)     ;       LSTK(1)=N
3099 002054 013737 003140' 005110'      MOV     RESPTR,OPTMP9   ;       NEX=1
3100 002062 011001                      MOV     (R0),R1         ;20     T=LSTK(LPTR)
3101 002064 002002                      BGE     7$              ;       IF T<0 THEN
3102 002066 005301                      DEC     R1              ;                       T=T-1
3103 002070 005401                      NEG     R1              ;       ABS(T)
3104 002072 006201                      ASR     R1              ;       T=.5*ABS(T)
```

```
3105 002074 010140                      MOV     R1,-(R0)        ;       LPTR=LPTR+1, LSTK(LPTR)=T
3106 002076 020127 000001               CMP     R1,#1           ;       IF T .NE. 1 THEN
3107 002102 001367                      BNE     6$              ;               GO TO 20
3108 002104 062737 000002 005106'       ADD     #2,OPTMP8       ;       HPTR=HPTR+1
3109 002112 013777 005110' 005106'      MOV     OPTMP9,@OPTMP8  ;       HSTK(HPTR)=NEX
3110 002120 013777 005110' 005110'      MOV     OPTMP9,@OPTMP9  ;       P(NEX)=NEX
3111 002126 062737 000002 005110'       ADD     #2,OPTMP9       ;       NEX=NEX+1
3112 002134 005720              8$:     TST     (R0)+           ;30     LPTR=LPTR-1
3113 002136 005410                      NEG     (R0)            ;       LSTK(LPTR)=-LSTK(LPTR)
3114 002140 002750                      BLT     6$              ;       IF LSTK(LPTR)<0 THEN GO TO 20
3115                                                             ;       HSTK(HPTR) = HSTK(HPTR) M2 HSTK(HPTR-1)
3116 002142 017703 005106'              MOV     @OPTMP8,R3      ;GET J
3117 002146 162737 000002 005106'       SUB     #2,OPTMP8       ;POP HSTK
3118 002154 017701 005106'              MOV     @OPTMP8,R1      ;GET I
3119 002160                             CALL    GDCMP           ;       IF A(I)<A(J) THEN GO TO 10
3120 002164 000404                      BR      9$              ;<
3121 002166 000403                      BR      9$              ;=
3122 002170 010377 005106'              MOV     R3,@OPTMP8      ;> 5    M2=J
3123 002174 000410                      BR      11$             ;       GO TO 30
3124 002176 010177 005106'      9$:     MOV     R1,@OPTMP8      ;10     M2=I
3125 002202 000414                      BR      13$             ;       GO TO 50
3126 002204                     10$:    CALL    GDCMP           ;20     IF A(I)<A(J) THEN GO TO 40
3127 002210 000410                      BR      12$             ;<
3128 002212 000407                      BR      12$             ;=
3129 002214 010312                      MOV     R3,(R2)         ;>      P(T)=J
3130 002216 010302              11$:    MOV     R3,R2           ;30     T=J
3131 002220 011203                      MOV     (R2),R3         ;       J=P(T)
3132 002222 020203                      CMP     R2,R3           ;       IF T .NE. J THEN
3133 002224 001367                      BNE     10$             ;               GO TO 20
3134 002226 010112                      MOV     R1,(R2)         ;       P(T)=I
3135 002230 000406                      BR      14$             ;       EXIT
3136 002232 010112              12$:    MOV     R1,(R2)         ;40     P(T)=I
3137 002234 010102              13$:    MOV     R1,R2           ;50     T=I
3138 002236 011201                      MOV     (R2),R1         ;       I=P(T)
3139 002240 020201                      CMP     R2,R1           ;       IF T .NE. I THEN
3140 002242 001360                      BNE     10$             ;               GO TO 20
3141 002244 010312                      MOV     R3,(R2)         ;       P(T)=J
3142
3143 002246 020037 005104'      14$:    CMP     R0,OPTMP7       ;       IF LPTR>1 THEN
3144 002252 103730                      BLO     8$              ;               GO TO 30
3145 002254 017700 005106'              MOV     @OPTMP8,R0      ;       I=HSTK(HPTR)
3146
3147 002260 013701 005070'              MOV     OPTMP1,R1       ;       DO 40 K=1, N
3148 002264 005002                      CLR     R2              ;TRACE THE CHAIN AND REPLACE WITH INDICES
3149 002266 005302              15$:    DEC     R2              ;       -K
3150 002270 011003                      MOV     (R0),R3         ;       T=P(I)
3151 002272 010210                      MOV     R2,(R0)         ;       P(I)=-K
3152 002274 010300                      MOV     R3,R0           ;       I=T
3153 002276                             SOB     R1,15$
3154
3155                            ;+
3156                            ;NOW INVERT THE PERMUTATION VECTOR AND CORRECT THE ORIGIN, ALL PREVIOUS
3157                            ;  OPERATIONS WERE IN ORIGIN ONE.
3158                            ;-
3159
3160 002302 013700 005070'              MOV     OPTMP1,R0       ;       DO 30 I=N, 1, -1
3161 002306 010001              16$:    MOV     R0,R1           ;       L=I
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 63-3
MIXED FUNCTION          GRADE UP

 3162 002310  163701  005100'                SUB     OPTMP5,R1       ;       L=I-NOTORG  (ORIGIN CORRECTION)
 3163 002314  010004                         MOV     R0,R4           ;       J=P(I)
 3164 002316  005304                         DEC     R4
 3165 002320  006304                         ASL     R4
 3166 002322  063704  003140'                ADD     RESPTR,R4
 3167 002326  011402                         MOV     (R4),R2
 3168 002330  002015                         BGE     18$             ;       IF J .GE 0 THEN GO TO 30
 3169 002332  005402                 17$:    NEG     R2              ;20     J=-J
 3170 002334  010204                         MOV     R2,R4           ;       K=P(J)
 3171 002336  005304                         DEC     R4
 3172 002340  006304                         ASL     R4
 3173 002342  063704  003140'                ADD     RESPTR,R4
 3174 002346  011403                         MOV     (R4),R3
 3175 002350  010114                         MOV     R1,(R4)         ;       P(J)=L
 3176 002352  010201                         MOV     R2,R1           ;       L=J
 3177 002354  163701  005100'                SUB     OPTMP5,R1       ;       L=J-NOTORG  (ORIGIN CORRECTION)
 3178 002360  010302                         MOV     R3,R2           ;       J=K
 3179 002362  002763                         BLT     17$             ;       IF J<0 THEN GO TO 20
 3180 002364                          18$:    SOB     R0,16$          ;30     CONTINUE
 3181
 3182                                 ;+
 3183                                 ;DONE, NOW CLEAN UP AND RETURN
 3184                                 ;-
 3185 002370  062706  000100                 ADD     #100,SP         ;TRASH THE SORT STACKS
 3186 002374                                 RETURN
 3187
 3188                                 ;+
 3189                                 ; COMPARE ROUTINE FOR SORT
 3190                                 ;-
 3191
 3192 002376  062716  000002         GDCMP:  ADD     #2,(SP)         ;EQUAL RETURN ASSUMED
 3193 002402                                 PUSH    <R1,R2,R3,R4>   ;SAVE SOME REGISTERS
 3194 002412  163701  003140'                SUB     RESPTR,R1       ;CALCULATE POINTER
 3195 002416  006201                         ASR     R1              ;INTO A FROM
 3196 002420                                 MUL     OPTMP2,R1       ;POINTER I
 3197 002434  063701  003122'                ADD     MONPTR,R1       ;=MONPTR+STRIDE*(RESPTR-I)/2
 3198 002440  010102                         MOV     R1,R2           ;MOVE OFFSET TO CORRECT REG
 3199 002442  163703  003140'                SUB     RESPTR,R3       ;CALCULATE POINTER
 3200 002446  006203                         ASR     R3              ;INTO A FROM
 3201 002450                                 MUL     OPTMP2,R3       ;POINTER J
 3202 002464  063703  003122'                ADD     MONPTR,R3       ;=MONPTR+STRIDE*(RESPTR-J)/2
 3203 002470  013704  005074'                MOV     OPTMP3,R4       ;NUMBER OF ITEMS TO SCAN
 3204 002474                          1$:    CALL    @LARGCV         ;GET LEFT ARGUMENT
 3205 002500                                 CALL    @RARGCV         ;GET RIGHT ARGUMENT
 3206 002504                                 CALL    @COMRTN         ;COMPARE THEM
 3207 002510  002410                         BLT     3$              ;<
 3208 002512  003013                         BGT     4$              ;>
 3209 002514                                 SOB     R4,1$           ;=, KEEP LOOKING
 3210 002520                          2$:    POP     <R4,R3,R2,R1>   ;ALL EQUAL RETURN
 3211 002530                                 RETURN
 3212
 3213 002532  163766  005102' 000010  3$:    SUB     OPTMP6,10(SP)   ;LESS THAN RETURN
 3214 002540  000767                         BR      2$
 3215
 3216 002542  063766  005102' 000010  4$:    ADD     OPTMP6,10(SP)   ;GREATER THAN RETURN
 3217 002550  000763                         BR      2$
 3219 002552                                 LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 64
MIXED FUNCTION          GRADE UP
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 64
MIXED FUNCTION          GRADE UP

   3221 002552                          MOPINS  MMD,122
   3316 002552                                  LIST
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 65
MIXED FUNCTION          GRADE UP
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 65
MIXED FUNCTION          GRADE UP

   3318 002552                          DOPINS  DMD,122
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 66
MIXED FUNCTION          GRADE UP
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 66
MIXED FUNCTION          GRADE UP

   3864 002552                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 67
MIXED FUNCTION          GRADE UP
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 67
MIXED FUNCTION          GRADE UP

   3?56 002552                          DOPINS  MEM,70
   3921 002552                                  LIST
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 68
MIXED FUNCTION          GRADE UP

   3923 002552                          MOPINS  NUB,106
   3986 002552                                  LIST
   3987
   3988
   3989
   3990
   3991
   3992 002552                          DOPINS  UNON,106
   4022 002552                                  LIST
   4023
   4024
   4025
   4026
   4027
   4028 002552                          DOPINS  INTR,66
   4042 002552                                  LIST
   4043
   4044
   4045
   4046
   4047
   4048 002552                          DOPINS  SDIF,50
   4161 002552                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 69
MIXED FUNCTION          GRADE UP
```

```
4163 002552                          DOPINS  SCON,234
4179 002552                                  LIST
4180
4181
4182 002552                          DOPINS  CONT,112
4200 002552                                  LIST
4201
4202
4203 002552                          DOPINS  PSUB,236
4209 002552                                  LIST
4210
4211
4212 002552                          DOPINS  SUBS,116
4288 002552                                  LIST
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 70
MIXED FUNCTION          GRADE UP

   4290 002552                          MOPINS  EXE,70
   4336 002552                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 71
MIXED FUNCTION          GRADE UP
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 71
MIXED FUNCTION          GRADE UP

   4338 002552                          MOPINS  ENCD,76
   4411 002552                                  LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 72
MIXED FUNCTION          GRADE UP
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 72
MIXED FUNCTION          GRADE UP

   4413 002552                          MOPINS  FMT,160
   4423 002552                                  LIST
   4424
   4425
   4426
   4427
   4428
   4429 002552                          DOPINS  FMT,160
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 73
MIXED FUNCTION          GRADE UP
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 73
MIXED FUNCTION          GRADE UP

   4953 002552                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 74
MIXED FUNCTION          GRADE UP
```

```
 4955 002552                    MOPINS  IBM,132,,,MONIBM
 4956                           .SBTTL  MIXED FUNCTION          MONADIC I BEAM
 4958 002552                            LIST
 4959
 4960
 4961 002552                    DOPINS  IBM,132
 5046 002552                            LIST
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 75
MIXED FUNCTION          MONADIC I BEAM

 5048 002552                            MOPINS  QUAD,74,T.NIL+T.SPOP
 5075 002552                                    LIST
 5076
 5077
 5078
 5079
 5080
 5081 002552                            DOPINS  QUAD,74,T.NIL+T.SPOP
 5097 002552                                    LIST
 5098
 5099
 5100
 5101
 5102
 5103 002552                            MOPINS  QQAD,142,T.NIL+T.SPOP
 5135 002552                                    LIST
 5136
 5137
 5138
 5139
 5140
 5141 002552                            DOPINS  QQAD,142,T.NIL+T.SPOP
 5152 002552                                    LIST
 5153
 5154
 5155
 5156
 5157
 5158 002552                            MOPINS  DQAD,134,T.NIL+T.SPOP
 5192 002552                                    LIST
 5193
 5194
 5195
 5196 002552                            DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
 5199 002552                                    LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 76
MIXED FUNCTION          MONADIC I BEAM
```

```
5201 002552                          MOPINS  BXAV,162,T.NIL
5219 002552                                  LIST
5220
5221
5222
5223
5224
5225 002552                          MOPINS  BXCT,164,T.NIL+T.SPOP
5234 002552                                  LIST
5235
5236
5237 002552                          DOPINS  BXCT,164,T.NIL+T.SPOP
5272 002552                                  LIST
5273
5274
5275
5276
5277
5278 002552                          MOPINS  BXIO,166,T.NIL+T.SPOP
5287 002552                                  LIST
5288
5289
5290 002552                          DOPINS  BXIO,166,T.NIL+T.SPOP
5302 002552                                  LIST
5303
5304
5305
5306
5307 002552                          MOPINS  BXLC,212,T.NIL
5308                                 .SBTTL  SYSTEM VARIABLE         LINE COUNTER
5309
5310 002552  013703  003314'                MOV     SIFRAM,R3               ;Point to current frame
5311 002556  013763  004764'  000002        MOV     FLAGS,F.FLAG(R3)        ;Save current flags (Just in case)
5312 002564  005002                         CLR     R2                      ;Clear count of F-Frames
5313 002566  032763  001000  000002  1$:    BIT     #F$FUNC,F.FLAG(R3)      ;IS THIS AN F-FRAME?
5314 002574  001401                         BEQ     2$                      ;No, don't count it
5315 002576  005202                         INC     R2                      ;Increment count of F-Frames
5316 002600  011303                  2$:    MOV     @R3,R3                  ;POINT TO NEXT FRAME
5317 002602  001371                         BNE     1$                      ;IF THERE IS ONE
5318 002604  010200                         MOV     R2,R0                   ;Copy count
5319 002606  006300                         ASL     R0                      ;(in words)
5320 002610  062700  000010                ADD     #A.DIM1+2,R0            ;Add in overhead
5321 002614                                 CALL    GETFRA                  ;Build result frame
5322 002620  062720  000010                ADD     #A.DIM1+2,(R0)+         ;Point to data
5323 002624  012720  102001                MOV     #<A.ASB+.INTEGER>*400+1,(R0)+  ;Integer vector
5324 002630  010220                        MOV     R2,(R0)+                ;A.ALEN
5325 002632  010220                        MOV     R2,(R0)+                ;A.DIM1
5326 002634  001417                        BEQ     7$                      ;No suspended functions
5327
5328 002636  013703  003314'               MOV     SIFRAM,R3               ;POINT TO CURRENT FRAME
5329 002642  032763  001000  000002  4$:    BIT     #F$FUNC,F.FLAG(R3)      ;IS THIS AN F-FRAME?
5330 002650  001003                         BNE     6$                      ;YES!
5331 002652  011303                  5$:    MOV     @R3,R3                  ;POINT TO NEXT FRAME
5332 002654  001372                         BNE     4$                      ;IF THERE IS ONE
5333 002656  104402                         SYSERR                          ;Cannot happen!
5334
5335 002660  016301  000012          6$:    MOV     F.LINE(R3),R1           ;POINT TO CURRENT LINE
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 76-1
SYSTEM VARIABLE          LINE COUNTER

 5336 002664  016120  000010                  MOV     R.LNUM(R1),(R0)+        ;GET LINE NUMBER
 5337 002670                                  SOB     R2,5$                   ;AND CONTINUE IF RESULT NOT FULL
 5338 002674                          7$:     RETURN                          ;ELSE WE'RE DONE
 5340 002676                                  LIST
 5341
 5342
 5343
 5344
 5345
 5346
 5347 002676                          MOPINS  BXPP,170,T.NIL+T.SPOP
 5356 002676                                  LIST
 5357
 5358
 5359 002676                          DOPINS  BXPP,170,T.NIL+T.SPOP
 5381 002676                                  LIST
 5382
 5383
 5384
 5385
 5386
 5387
 5388 002676                          MOPINS  BXPW,172,T.NIL+T.SPOP
 5397 002676                                  LIST
 5398
 5399
 5400 002676                          DOPINS  BXPW,172,T.NIL+T.SPOP
 5417 002676                                  LIST
 5418
 5419
 5420
 5421
 5422
 5423 002676                          MOPINS  BXRL,174,T.NIL+T.SPOP
 5432 002676                                  LIST
 5433
 5434
 5435 002676                          DOPINS  BXRL,174,T.NIL+T.SPOP
 5447 002676                                  LIST
 5448
 5449
 5450
 5451
 5452
 5453 002676                          MOPINS  BXTS,200,T.NIL
 5562 002676                                  LIST
 5563
 5564
 5565
 5566
 5567
 5568
 5569 002676                          MOPINS  BXWA,176,T.NIL
 5592 002676                                  LIST
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 77
SYSTEM VARIABLE          LINE COUNTER
```

L 15

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 77
SYSTEM VARIABLE          LINE COUNTER

   5594 002676                              MOPINS  BXCR,206
   5716 002676                                      LIST
```

M 15

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 78
SYSTEM VARIABLE          LINE COUNTER

   5718 002676                              MOPINS  BXEX,202
```

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 78
SYSTEM VARIABLE LINE COUNTER

```
 5718 002676                          MOPINS  BXEX,202
 5775 002676                                  LIST
```


APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 79
SYSTEM VARIABLE LINE COUNTER

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 79
SYSTEM VARIABLE          LINE COUNTER

   5777 002676                              MOPINS  BXFX,210
   5879 002676                                      LIST
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 80
SYSTEM VARIABLE          LINE COUNTER

   5881 002676                          MOPINS  BXNC,204
   5950 002676                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 81
SYSTEM VARIABLE          LINE COUNTER

   5953 002674                          MOPINS  BYNL,214
```

C 16

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 81
SYSTEM VARIABLE          LINE COUNTER

   5952 002676                          MOPINS  BXNL,214
   5963 002676                                  LIST
   5964
   5965
   5966 002676                          DOPINS  BXNL,214
   6104 002676                                  LIST
```

D 16

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 82
SYSTEM VARIABLE          LINE COUNTER

   6106 002676                          DOPI.>  EGET 150,I,DIM
```

```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 82
SYSTEM VARIABLE          LINE COUNTER

   6106 002676                          DOPI..S  FGET,150,T.DIM
   6330 002676                                   LIST
   6331
   6332
   6333
   6334
   6335
   6336 002676                          DOPINS   FPUT,152,T.DIM
   6485 002676                                   LIST
```



```
APLOO3  MACRO M1113  08-FEB-80 09:22  PAGE 83
SYSTEM VARIABLE          LINE COUNTER

   6487 002676                          DOPINS   FPNT,154,T.DIM
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 83
SYSTEM VARIABLE          LINE COUNTER

   6487 002676                              DOPINS  FPNT,154,T.DIM
   6714 002676                                      LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 84
SYSTEM VARIABLE          LINE COUNTER

   6716 002676                              MOPINS  1SCN,130
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 84
SYSTEM VARIABLE          LINE COUNTER

   6716 002676                          MOPINS  1SCN,130
   6723 002676                                  LIST
   6724
   6725
   6726 002676                          MOPINS  SCAN,62,T.DIM
   6732 002676                                  LIST
   6733
   6734
   6735 002676                          MOPINS  1RDT,124
   6741 002676                                  LIST
   6742
   6743
   6744 002676                          MOPINS  RDCT,60,T.DIM
   7044 002676                                  LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 85
SYSTEM VARIABLE          LINE COUNTER

   7046 002676                          MOPINS  GETIDN,0
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 85
SYSTEM VARIABLE          LINE COUNTER

   7046 002676                              MOPINS  GETIDN,0
   7193 002676                                      LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 86
SYSTEM VARIABLE          LINE COUNTER

   7195 002676                              MOPINS  SATAN,0
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 86
SYSTEM VARIABLE          LINE COUNTER

 7195 002676                              MOPINS  SATAN,0
 7404 002676                                      LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 87
SYSTEM VARIABLE          LINE COUNTER

 7406 002676                              MOPINS  OUTR.156
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 87
SYSTEM VARIABLE          LINE COUNTER

   7406 002676                              MOPINS  OUTR,156
   7476 002676                                      LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 88
SYSTEM VARIABLE          LINE COUNTER

   7478 002676                              MOPINS  DOUTER,0
```

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 88
SYSTEM VARIABLE          LINE COUNTER

 7478 002676                            MOPINS  DOUTER,0
 7504 002676                                    LIST
 7505
 7506
 7507
 7508
 7509
 7510 002676                            MOPINS  PRDSET,0
 7564 002676                                    LIST
 7565
 7566
 7567
 7568
 7569
 7570 002676                            MOPINS  INSPAC,0
 7612 002676                                    LIST
```



```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 89
SYSTEM VARIABLE          LINE COUNTER

 7614 002676                            DOPINS  INNR,156
```

K 16

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 89
SYSTEM VARIABLE          LINE COUNTER

   7614 002676                          DOPINS  INNR,156
```

L 16

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 90
SYSTEM VARIABLE          LINE COUNTER

   7960 002676                          LIST
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 90
SYSTEM VARIABLE          LINE COUNTER

   7960 002676                                    LIST
```



```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 91
SYSTEM VARIABLE          LINE COUNTER

   7962 002676                              MOPINS  SUBR,0
```

```
7962 002676                          MOPINS  SUBR,0
```

B 1
C 1
D 1
J 5
K 5
L 5
E 10
F 10
G 10
M 14
N 14
B 15

B 1
C 1
D 1
E 1
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1
B 2
C 2
D 2
E 2
F 2
G 2
H 2
I 2
J 2
K 2
L 2
M 2
N 2
B 3
C 3
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5
I 5

J 5
K 5
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7
N 7
B 8
C 8
D 8
E 8
F 8
G 8
H 8
I 8
J 8
K 8
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9
J 9
K 9
L 9
M 9
N 9
B 10
C 10
D 10

APL002

E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14

APL003

M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 92
SYSTEM VARIABLE   LINE COUNTER

```
   8725 002676                                    LIST
```

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 93
SYSTEM VARIABLE          LINE COUNTER

   8727 002676                              MOPINS  RAND,0
   8771 002676                                      LIST
```

D 1

```
APL003  MACRO M1113  08-FEB-80 09:22  PAGE 94
SYSTEM VARIABLE          LINE COUNTER

      1         000001                          .END
```

E 1

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    C.BLOK= 000064    D.COMB= 000001    D.1ROT= 000003
AC1    =%000001          CHRLBR= 000073    C.BLSZ= 000140    D.COMP= 000001    ECOMCT= ****** GX
AC2    =%000002          CHRLPR= 000072    C.BUFF= 177760    D.CONT= 000006    EDICHR  005063R   025
AC3    =%000003          CHRMNS= 000137    C.BUFL= 177762    D.CTLM= 000000    EDICNT  005064R   025
AC4    =%000004          CHRMUL= 000075    C.CCNT= 177766    D.DEAL= 000003    EDIFLG  005062R   025
AC5    =%000005          CHRNEG= 000100    C.CPTR= 177764    D.DECD= 000002    EDIT   = 000003
ALLINT  003274R   025    CHRPDL= 000271    C.CZ   = 001000   D.DIV = 000000    EIS    = 000000
ARYRET  003276R   025    CHRPLU= 000055    C.DIG  = 000040   D.DMD = 000010    ELESIZ  005132R   025
ASCQUO= 000042           CHRPWR= 000120    C.FASW= 020000    D.DQAD= 177777    ENDCOR  000004R   025
A.ALEN= 000004           CHRQST= 000121    C.FLEN= 000010    D.DROP= 000002    EOF    = 104475
A.ARG  = 000040          CHRQUO= 000113    C.FLSH= 004000    D.ENCD= 000002    ERRMAX= 000166
A.ASB  = 000200          CHRRBR= 000047    C.FRC  = 000001   D.EQ   = 000000    ERRNUM  004772R   025
A.BITS= 000003           CHRRPR= 000042    C.IDEN= 000010    D.EXPD= 000002    ERRPTR  004770R   025
A.CON  = 000020          CHRSEM= 000074    C.NAME= 000102    D.FGET= 000005    ERR0    004774R   025
A.DIMS= 000002           CHRSMO= 000112    C.NOBF= 000400    D.FMT  = 000004   EXEBUF  005054R   025
A.DIM1= 000006           CHRSPA= 000040    C.NUM  = 000020   D.FPNT= 000005    E$CT    005056R   025
A.NUM  = 000006          CHRUPA= 000131    C.OP   = 000004   D.FPUT= 000005    E$PT    005060R   025
A.PTR  = 000000          CHR.BS= 000010    C.OPN  = 010000   D.GE   = 000000    FA.APD= 000100
A.TMP  = 000010          CHR.CC= 000003    C.RDLN= 177776    D.GT   = 000000    FA.CRE= 000010
BADDIR= 104463           CHR.CR= 000015    C.REWT= 002000    D.IBM  = 000002    FA.DLK= 001000
BADERR= 104437           CHR.CZ= 000032    C.RND  = 000010   D.IND  = 000002    FA.ENB= 100000
BLZBUB  005116R   025    CHR.GT= 000046    C.SBIT= 177756    D.INNR= 000013    FA.EXC= 002000
BRSELF  005046R   025    CHR.LF= 000012    C.SPFX= 000100    D.INTR= 000006    FA.EXT= 000004
CARVE  = ****** GX       CHR.LT= 000043    C.SPGL= 000200    D.LE   = 000000    FA.NSP= 000100
CCNT    005072R   025    CHR.SI= 000017    C.SPLX= 000001    D.LOG  = 000001    FA.NSR= 000200
CHANEX  003060R   025    CHR.SL= 000057    C.SPRS= 000002    D.LT   = 000000    FA.POS= 010000
CHAN00  000253R   025    CHR.SO= 000016    C.SRC  = 177774   D.MAX  = 000000    FA.RD  = 000001
CHAN01  000330R   025    CHR.TB= 000011    C.SRCP= 000024    D.MEM  = 000002    FA.RWD= 004000
CHAN02  000512R   025    CHR.UL= 000106    C.TT   = 000004   D.MIN  = 000000    FA.SEQ= 040000
CHAN03  000674R   025    CHR..A= 000141    C.WIDC= 177770    D.MUL  = 000000    FA.SHR= 000040
CHAN04  001056R   025    CHR..E= 000145    C.WIDR= 177772    D.NE   = 000000    FA.TMP= 000020
CHAN05  001240R   025    CHR..S= 000163    DEBUG  = 000000   D.NN   = 000000    FA.WCK= 020000
CHAN06  001422R   025    CHR..T= 000164    DEFERR= 104405    D.NR   = 000000    FA.WRT= 000002
CHAN07  001604R   025    CHR..U= 000165    DEVIL   005114R   025 D.OR   = 000000    FD.BLK= 000010
CHAN08  001766R   025    CHR..Z= 000172    DEVNTF= 104470    D.PLUS= 000000    FD.CCL= 000002
CHAN09  002150R   025    CHR..0= 000060    DFXDIM= ****** GX D.PSUB= 000006    FD.COM= 020000
CHAN10  002332R   025    CHR..1= 000061    DIGITS  000006R   025 D.PWR  = 000001    FD.CR  = 000002
CHAN11  002514R   025    CH.AND= 000001    DIMAX  = 000020   D.QQAD= 000000    FD.DIR= 000010
CHAN12  002676R   025    CMDBUF  003250R   025 DIMBLK  003112R   025 D.QUAD= 000000    FD.FTN= 000001
CHAN13  000022R   023    CMDMAX= 000010    DIMPTR  003114R   025 D.RES  = 000000    FD.F11= 040000
CHAN14  000204R   023    CNTSIZ  005130R   025 DIMTYP  003116R   025 D.RHO  = 000000    FD.INS= 000010
CHNOPN= 104471           COMBIN  005076R   025 DIRFUL= 104466    D.ROT  = 000003    FD.ISP= 002000
CHRAND= 000051           COMCNT  005126R   025 DOMERR= 104417    D.SCON= 000006    FD.MNT= 100000
CHRBEL= 000007           COMERR= 104425    DPTERR= 104416    D.SDIF= 000006    FD.OSP= 004000
CHRBKR= 000133           COMPTR  005122R   025 DYDCNT  003224R   025 D.SUB  = 000000    FD.PLC= 000004
CHRBOL= 000200           COMRTN  003156R   025 D$DEAL  000000RG   020 D.SUBS= 000006    FD.PRN= 000004
CHRBOX= 000114           COMSKP  005124R   025 D$RES  = ****** GX D.TAKE= 000002    FD.PSE= 010000
CHRCMA= 000054           CORPTR  005120R   025 D$ROT   001034RG   020 D.TRAN= 000004    FD.RAH= 000001
CHRCOL= 000076           CS.ASS= 000200    D$1ROT  001014RG   020 D.UNON= 000006    FD.RAN= 000002
CHRDEL= 000107           CS.BOO= 000010    D.AND  = 000000   D.$MAX= 000000    FD.REC= 000001
CHRDIV= 000053           CS.FLO= 000020    D.BXCT= 000007    D.$MIN= 000000    FD.RWM= 000001
CHRDLR= 000176           CS.IDN= 000040    D.BXIO= 000007    D.$MUL= 000000    FD.SDI= 000020
CHRDLT= 000110           CS.ONE= 000100    D.BXNL= 000007    D.$PLU= 000000    FD.SQD= 000040
CHRDOT= 000056           CTEPS   000152R   025 D.BXPP= 000007    D.$RES= 000000    FD.TTY= 000004
CHREP  = 000105          CTMAX   000212R   025 D.BXPW= 000007    D.$SUB= 000000    FD.WBH= 000002
CHRERR= 104431           CTRMAX  000172R   025 D.BXRL= 000007    D.1COM= 000001    FF.CHR= 000005
CHRFIL= 000177           C.ALOC= 000162    D.CIRC= 000001    D.1EXP= 000002    FF.NV  = 000003
```

APLO03 MACRO M1113 08-FEB-80 09:22 PAGE 94-2
SYMBOL TABLE

```
FF.POE= 000002          F.EFN = 000050          F.URBD= 000020          LININC  005026R 025 M.FIL = 000177
FF.RWD= 000001          F.END = 002000          F.VAL = 177776          LISIZE= 000020      M.FLR = 177777
FF.RWF= 000006          F.EOBB= 000032          F.VBN = 000064          LITOP   005072R 025 M.FMT = 000004
FF.SPC= 000004          F.ERR = 000052          F.VBSZ= 000060          LNLNG = 104541      M.GDUP= 000003
FHEADR  005012R 025     F.FACC= 000043          GDCMP   002376R 020     LOCLCT  005036R 025 M.GDWN= 000003
FHRET   005040R 025     F.FFBY= 000014          GETCO1= ****** GX       LOWCOR  000002R 025 M.GETI= 000006
FILNTF= 104467          F.FLAG= 000002          GETFRA= ****** GX       LSTERP  004766R 025 M.IBM = 000003
FIS   = 000000          F.FLB = 000100          GETREG= ****** GX       LWIDTH  000070R 025 M.IDN = 000000
FISCON= 046000          F.FNAM= 000110          GNOME   003102R 025     L.BITS= 000003      M.INSP= 000006
FIXDIM= ****** GX       F.FNB = 000102          HASHC   005136R 025     L.CNT = 000002      M.IOTA= 000000
FIXPTR  003074R 025     F.FNDF= 000010          HASHTB  005140R 025     L.PRNT= 000001      M.LIT = 000205
FLAGS   004764R 025     F.FTYP= 000116          HSHMSK= 177601          L.PTR = 000000      M.LN  = 000001
FLCBUF  003250R 025     F.FUN = 000010          HSHSIZ= 000200          L.PTR1= 000004      M.MMD = 000010
FLINE   005014R 025     F.FVER= 000120          IASH  = ****** GX       L.SUBS= 000002      M.NEG = 000000
FLOEXP  003270R 025     F.HEAD= 000006          IASHC = ****** GX       MACHIN= 000106      M.NOT = 000000
FLONEG  003272R 025     F.HIBK= 000004          ICHAN   005050R 025     MAXCHN= 000014      M.NUB = 000006
FLSWTB  000222R 025     F.LBL = 000020          ILARGC= ****** GX       MFXDIM= ****** GX   M.NUM = 000204
FLTLEN= 000010          F.LINE= 000012          ILFNAM= 104464          MONBLK  003120R 025 M.OUTR= 000012
FNINSI  005044R 025     F.LOC = 000004          ILLCHN= 104540          MONCVT  003132R 025 M.PITM= 000000
FORIGN  000032R 025     F.LSTC= 000004          ILLSW = 104565          MONDIM= ****** GX   M.PRDS= 000006
FO.APD= 000106          F.LUN = 000042          IMUL  = ****** GX       MONLEN  003130R 025 M.QQAD= 000000
FO.MFY= 000002          F.MBCT= 000054          INDERR= 104410          MONPTR  003122R 025 M.QUAD= 000000
FO.RD = 000001          F.MBC1= 000055          INNTMP  003250R 025     MONSFT  003126R 025 M.RAND= 000012
FO.UPD= 000006          F.MBFG= 000056          INNTM2  005016R 025     MONTYP  003124R 025 M.RAVL= 000000
FO.WRT= 000016          F.MON = 000002          INPTR   000064R 025     MONVAL  003134R 025 M.RCIP= 000000
FPMODE  000042R 025     F.NIL = 000001          INSNAM= 104462          MOVROT  001414R 020 M.RDCT= 000011
FPP   = 000000          F.NRBD= 000024          IOERR = 104477          M$BXLC  002552RG 020 M.REVS= 000003
FPPLEN= 000004          F.NREC= 000030          IRARGC= ****** GX       M$GDUP  001472RG 020 M.ROLL= 000000
FRARGC= ****** GX       F.OPN = 000010          I.CNT = 177776          M$GDWN  001460RG 020 M.SATA= 000001
FREBLK  007400R 025     F.OVBS= 000030          I.DIM = 177772          M$IBM   002552RG 020 M.SCAN= 000011
FRESCT= ****** GX       F.PREV= 000000          I.LEN = 000012          M$REVS  000702RG 020 M.SHAP= 000000
FUZZ    000142R 025     F.PRNT= 100000          I.OERR= 000062          M$1REV  000656RG 020 M.SIGN= 000000
FXPTR1  003076R 025     F.PROT= 010000          I.OLD = 177770          M.ABS = 000000      M.SUBR= 000001
F$EVAL= 000100          F.RACC= 000016          I.PTR = 177774          M.BOL = 000200      M.SYM = 000203
F$EVI = 000200          F.RATT= 000001          I.STR = 177766          M.BXAV= 000007      M.SYN = 000201
F$EXEC= 000400          F.RCNM= 000034          LABERR= 104406          M.BXCR= 000007      M.TRAN= 000012
F$FUNC= 001000          F.RCTL= 000017          LABLPT  005042R 025     M.BXCT= 000007      M.$ABS= 000000
F.ACTL= 000076          F.READ= 004000          LARGBL  003162R 025     M.BXEX= 000007      M.$CEI= 000000
F.ALOC= 000040          F.REDF= 000040          LARGCT= ****** GX       M.BXFX= 000007      M.$FLR= 000000
F.BBFS= 000062          F.RES = 000040          LARGCV  003176R 025     M.BXIO= 000007      M.$IDN= 000000
F.BDB = 000070          F.RET = 177772          LARGFL  003160R 025     M.BXLC= 000003      M.$NEG= 000000
F.BGBC= 000057          F.RSIZ= 000002          LARGLE  003174R 025     M.BXNC= 000007      M.$SIG= 000000
F.BITS= 177774          F.RTYP= 000000          LARGLN  003166R 025     M.BXNL= 000007      M.1RDT= 000011
F.BKDN= 000026          F.R2  = 000014          LARGPT  003164R 025     M.BXPP= 000007      M.1REV= 000003
F.BKDS= 000020          F.R3  = 000016          LARGSF  003172R 025     M.BXPW= 000007      M.1SCN= 000011
F.BKEF= 000050          F.R4  = 000020          LARGTY  003170R 025     M.BXRL= 000007      NAMBUF  003240R 025
F.BKP1= 000051          F.SEQN= 000100          LARGVA  003200R 025     M.BXTS= 000007      NB.DEV= 000200
F.BKST= 000024          F.SPDV= 000072          LBINC   005100R 025     M.BXWA= 000007      NB.DIR= 000100
F.BKVB= 000064          F.SPUN= 000074          LBUFF   003316R 025     M.CEI = 177777      NB.NAM= 000004
F.BR  = 000020          F.STBK= 000036          LBUFLN= 000621          M.CHR = 000202      NB.SD1= 000400
F.CHR = 000075          F.STK = 000022          LCOUNT  005106R 025     M.CMT = 000206      NB.SD2= 001000
F.CNTG= 000034          F.STOP= 000004          LENERR= 104412          M.DOUT= 000006      NB.SNM= 000040
F.CODE= 000010          F.STP = 000001          LENTAB  000054R 025     M.DQAD= 000006      NB.STP= 000020
F.DFNB= 000046          F.SUSP= 000002          LEXPTR  003100R 025     M.ENCD= 000004      NB.SVR= 000010
F.DSPT= 000044          F.TO.I= ****** GX       LFTOP   005056R 025     M.EXE = 000000      NB.TYP= 000002
F.DVNM= 000134          F.TRC = 000002          LINC    005102R 025     M.EXP = 000001      NB.VER= 000001
F.EFBK= 000010          F.UNIT= 000136          LINE    005016R 025     M.FACT= 000001      ND.DEV= 000200
```

APL003 MACRO M1113 08-FEB-80 09:22 PAGE 94-3
SYMBOL TABLE

```
ND.EXT= 000002          OPTMP0  005066R     025 RINC    005104R     025 SYMTBF= 104404          T.DIM = 000100
ND.NAM= 000004          OPTMP1  005070R     025 RLTBAS  005066R     025 SYMTOP  005350R     025 T.FUN = 010000
ND.PPN= 000100          OPTMP2  005072R     025 ROTAT1  001040R     020 SYMTP   006272R     025 T.LIST= 000400
ND.$EX= 000000          OPTMP3  005074R     025 ROTINC  005070R     025 SYMWRN= 000054          T.LIT = 000020
ND.$E$= 000020          OPTMP4  005076R     025 ROUND   000010R     025 SYNERR= 104407          T.MON = 000010
ND.$NM= 000000          OPTMP5  005100R     025 ROVER   003312R     025 SYSER = ****** GX       T.NIL = 000004
ND.$N$= 000040          OPTMP6  005102R     025 RSTS  = 000000          SYSERR= 104402          T.NUM = 002000
ND.$PN= 001400          OPTMP7  005104R     025 RSX   = 000115          SYSTAT  000000R     024 T.OPFN= 000001
NEGSGN  005063R     025 OPTMP8  005106R     025 RTOP    005070R     025 SY.BAD= 000001          T.OTHR= 040000
NINPT1  003302R     025 OPTMP9  005110R     025 RT11  = 000000          SY.BRK= 100000          T.SOP = 000002
NINPT2  003300R     025 OPTM10  005112R     025 R.BITS= 000003          SY.BS = 000010          T.SPOP= 000040
NINSTK  003304R     025 OPTM14  005026R     025 R.FIX = 000001          SY.CLR= 000040          T.TMP = 000200
NINTMP  003310R     025 ORIGIN  000030R     025 R.LCT = 000002          SY.ERP= 000020          T.VAL = 100000
NINTYP  003306R     025 OUTPTR  000066R     025 R.LN  = 000002          SY.NOU= 000100          T.VAR = 001000
NOTAVL= 104472          OVROPR= 000003          R.LNUM= 000010          SY.OPN= 000002          UPPER = 177400
NOTOPN= 104473          PATCH   007344R     025 R.NEXT= 000006          SY.PMT= 001000          UTNAME  000344R     023
NPRT1   004762R     025 PRECSN  000052R     025 R.OFF = 000000          SY.SRT= 000400          VALERR= 104413
NULL    007364R     025 RANDAC  003064R     025 R.PREV= 000004          SY.SVC= 000004          VARBLE  007400R     025
NUMCHN= 000020          RANDY   003062R     025 R.PTR = 000000          SY.SVL= 000200          VARBLS  000000R     025
NUMDFL  003230R     025 RANERR= 104411          R.SEQ = 000003          S.AB  = 177776          VERNUM  000000R     025
NUMEFL  003233R     025 RARGBL  003204R     025 R.VAR = 000002          S.ASB = 100000          VRSION= 000002
NUMESN  003236R     025 RARGCT= ****** GX       R6    =%000006          S.BFHD= 000020          WIDTH   000072R     025
NUMFLG  003230R     025 RARGCV  003220R     025 R7    =%000007          S.BITS= 000002          WROUND  000020R     025
NUMLEN  000062R     025 RARGFL  003202R     025 SI    =%000005          S.CHR1= 000003          WSDATE  000136R     025
NUMNEG  003232R     025 RARGLE  003216R     025 SIDAMN= 104433          S.FATT= 000016          WSFULL= 104403
NUMQFL  003234R     025 RARGLN  003210R     025 SIFRAM  003314R     025 S.FDB = 000140          WSNAME  000074R     025
NUMRIG  003231R     025 RARGPT  003206R     025 SISTAK  006350R     025 S.FLB = 040000          WSTIME  000132R     025
NUMWID  003226R     025 RARGSF  003214R     025 SITOP   006346R     025 S.FNAM= 000006          $CFCC = ****** GX
NUMZFL  003235R     025 RARGTY  003212R     025 SITP    007144R     025 S.FNB = 000036          $CMPF = ****** GX
NYIERR= 104432          RARGVA  003222R     025 SIWARN= 000200          S.FNBW= 000017          $FIS  = ****** G
N.BLSZ= 000036          RBASE   005074R     025 SIZTAB  000044R     025 S.FNTY= 000004          $FPP  = ****** G
N.DESC= 000020          RCOUNR  005112R     025 SI.BIT= 177774          S.FTYP= 000002          $FPPLN= ****** G
N.DEV = 000032          RCOUNT  005110R     025 SI.LEN= 177772          S.LEN = 000002          $LDCIF= ****** GX
N.DEVN= 000034          REDCNT  005070R     025 SI.RET= 177772          S.LINK= 177774          $MACHN= ****** G
N.DID = 000024          REDIGN  005066R     025 SI.VAL= 177776          S.NFEN= 000020          $RSTS = ****** G
N.DVNM= 000032          REDOFF  005072R     025 SKPSIZ  005134R     025 S.TYPE= 140000          $RSX  = ****** G
N.EXT = 000014          REENTR  003060R     025 SORT    001500R     020 S.VAL = 000000          $RT11 = ****** G
N.FID = 000000          REGTYP= 000003          SPHOLD  000004R     024 TKDIMS  005072R     025 $STF  = ****** GX
N.FNAM= 000006          RESBIT  003142R     025 SPINIT  000002R     024 TKFRAM  005066R     025 .BOOLE= 000002
N.FTYP= 000014          RESBLK= ****** GX       STKINC= 000774          TKLEFT  005070R     025 .CHARA= 000000
N.FVER= 000016          RESCVT  003152R     025 STOP  = 104420          TMODE   000403R     023 .DEF. = 000002
N.NAME= 000006          RESERR= 104440          SUBCNT  003104R     025 TMPTAG= 000166          .FLOAT= 000006
N.NEXT= 000022          RESLEN  003150R     025 SUBDCT  003106R     025 TTYBUF  004140R     025 .FSRCB= ****** G
N.PPN = 000024          RESPTR  003140R     025 SUBPTR  003110R     025 TTYLEN= 000622          .INTEG= 000004
N.STAT= 000020          RESSFT  003146R     025 SYMB1   005344R     025 TURKEY= 104436          .MOLUN  000404R     023
N.UNIT= 000034          RESTYP  003144R     025 SYMF    005340R     025 TYPE  = 000166          .RAND = ****** GX
OBUFLN= 001000          RESULT  003136R     025 SYMPT1  005346R     025 T.BOL = 020000          ...GBL= 000000
OCHAN   005052R     025 RESVAL  003154R     025 SYMTAB  005352R     025 T.CON = 004000          ...TPC= 000020
OMODE   000402R     023 REVRS1  000706R     020

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
```

H 1

```
APLO03  MACRO M1113  08-FEB-80 09:22  PAGE 94-4
SYMBOL TABLE

DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    002676      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13604 WORDS  ( 54 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:21
APLO03,DM1:[305,301]APLO03/-SP=APLRX1,APLDEF,APLO03,APLOP,END
```

I 1

FILEID**APL004

```
  AAAAAA      PPPPPPPP      LL              000000        000000     44    44
  AAAAAA      PPPPPPPP      LL              000000        000000     44    44
AA      AA    PP      PP    LL            00    00    00      00    44    44
AA      AA    PP      PP    LL            00    00    00      00    44    44
AA      AA    PP      PP    LL            00    00    00    0000    44    44
AA      AA    PP      PP    LL            00    00    00    0000    44    44
AA      AA    PPPPPPPP      LL            00    00    00  00  00    4444444444
AA      AA    PPPPPPPP      LL            00    00    00  00  00    4444444444
AAAAAAAAAA    PP            LL            00    00    0000    00          44
AAAAAAAAAA    PP            LL            00    00    0000    00          44
AA      AA    PP            LL            00    00    00      00          44
AA      AA    PP            LL            00    00    00      00          44
AA      AA    PP            LLLLLLLLLL      000000        000000          44
AA      AA    PP            LLLLLLLLLL      000000        000000          44
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. M.. ..0 M1113 08-FEB-80 09:26
TABLE OF C.. ..ENTS


```
      2-     1    APLDEF
      2-     8    A P L
      2-     9
      2-    10    A PROGRAMMING LANGUAGE
      2-    11
      2-    12    STEVE POULSEN & DAVE VANN
      2-    13    11-Jul-76
      2-    14    ROBERT GOTTLIEB & PETER HAYNES
      2-    15    2-Feb-79
      2-    16
      2-    17    V2
      2-    18
      2-    19
      2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
      2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
      2-    22
      2-    23
      2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
      2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
      2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
      2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
      2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
      2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
            30    DEC.
      2-    31
      2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
      2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
      2-    34    CORPORATION.
      2-    35
      2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
      2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
      2-    38
      2-    39
      3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
      4-   139    DEFINITIONS     OPERATOR OVERLAYS
      5-   268    DEFINITIONS     REGISTERS
      6-   331    DEFINITIONS     MEMORY ORGANIZATION
      7-   377    DEFINITIONS     ORDERING PSECTS
      8-   426    DEFINITIONS     SI PARSE FRAMES
      9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
     10-   515    DEFINITIONS     SI EXECUTION FRAMES
     11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
     12-   607    DEFINITIONS     CHRTAB BITS
     12-   624    DEFINITIONS     SYSTEM STATUS BITS
     13-   6..    DEFINITIONS     F-FRAMES
     14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
     16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
     17-   869    DEFINITIONS     LIST STORAGE BLOCK
     18-   911    DEFINITIONS     INDEXING STACK FRAME
     19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
     20-   969    DEFINITIONS     CHANNEL BLOCK
     22-  1085    DEFINITIONS             RSX DATA
     23-  1110    DEFINITIONS     NAME BLOCKS
     33-  1612    DEFINITIONS     MACROS
     35-  1905    DEFINITIONS     APL CHARACTERS
     36-  1967    DEFINITIONS     ERRORS
     60-  2418    MIXED FUNCTION          DYADIC TRANSPOSE
```

APLC04 MACRO M1113 08-FEB-80 09:26
TABLE OF CONTENTS

71- 4339 MIXED FUNCTION MONADIC ENCODE (QUOTE)
72- 4414 MIXED FUNCTION MONADIC FORMAT
72- 4430 MIXED FUNCTION DYADIC FORMAT

APLDOV MACRO M1113 08-FEB-80 09:26 PAGE 1

```
                                        ;
                                        ;
                                        ;
```

APLO04 MACRO M1113 08-FEB-80 09:26 PAGE 2
APLDEF

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```


APL MACRO M1113 08-FEB-80 09:26 PAGE 42

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOO4  MACRO M1113  08-FEB-80 09:26  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APLOO4
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO4 IS:
      9                                 ;
     10                                 ;       APLOO4 < ...,APLOO4,APLOP
     11                                 ;
     12
     13          000004                OVROPR= 4                       ;OVERLAY SPECIFIED OPERATORS
```

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                                ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                                ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                                ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                                ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                                ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                                ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                                ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                                ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                                ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                                ;       INTACT!
                                ;--



                                ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                                ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                                ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                                ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                                ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                                ;MACROS.

                                .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-D.'NAME
                                        .PSECT  CODE
                                D$'NAME::
                                .ENDM

                                .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-M.'NAME
                                        .PSECT  CODE
                                        .IF NE NUM
                                M$'NAME::
                                    .ENDC
                                .ENDM

                                .MACRO  LIST
                                        .ENDC
                                .ENDM

                                .MACRO  PRMRES
                                        .IF EQ 1
                                .ENDM

                                        ;*********** ASSEMBLING THE FILE!!!!!
000000                          PRMRES
                                        LIST
```

DEFINITIONS ERRORS

```
201 007400                          DOPINS  PLUS,2,T.SOP,.INTEGER+CS.IDN+CS.ASS
208 007400                                  LIST
209
210 007400                          DOPINS  $PLU,-2
224 007400                                  LIST
225
226 007400                          DOPINS  SUB,4,T.SOP,.INTEGER+CS.IDN
233 007400                                  LIST
234
235 007400                          DOPINS  $SUB,-4
249 007400                                  LIST
250
251
252 007400                          DOPINS  MUL,6,T.SOP,.INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
264 007400                                  LIST
265
266
267 007400                          DOPINS  $MUL,-6
284 007400                                  LIST
285
286 007400                          DOPINS  DIV,10,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
323 007400                                  LIST
324
325
326 007400                          DOPINS  PWR,20,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
392 007400                                  LIST
393
394
395 007400                          DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
417 007400                                  LIST
418
419
420 007400                          DOPINS  CIRC,52,T.SOP,.FLOAT
477 007400                                  LIST
478
479
480 007400                          DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
490 007400                                  LIST
491
492 007400                          DOPINS  $MIN,-14
516 007400                                  LIST
517
518
519 007400                          DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
529 007400                                  LIST
530
531 007400                          DOPINS  $MAX,-16
555 007400                                  LIST
556
557
558 007400                          DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
589 007400                                  LIST
590
591
592 007400                          DOPINS  $RES,-12
616 007400                                  LIST
617
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 47-1
DEFINITIONS     ERRORS

    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

```
641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
780 007400                                  LIST
781
```

```
783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
786 007400                                  LIST
787
788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
791 007400                                  LIST
792
793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
799 007400                                  LIST
800
801
802 007400                          MOPINS  $IDN,-2
808 007400                                  LIST
809
810
811 007400                          MOPINS  $FLR,-14
855 007400                                  LIST
856
857
858 007400                          MOPINS  $CEI,-16
868 007400                                  LIST
869
870
871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
878 007400                                  LIST
879
880 007400                          MOPINS  $NEG,-4
887 007400                                  LIST
888
889
890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
901 007400                                  LIST
902
903 007400                          MOPINS  $SIG,-6
914 007400                                  LIST
915
916
917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
928 007400                                  LIST
929
930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
939 007400                                  LIST
940
941
942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
949 007400                                  LIST
950
951
952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
959 007400                                  LIST
960
961 007400                          PRMRES
966 007400                                  LIST
967
968
969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
977 007400                                  LIST
978
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

DEFINITIONS ERRORS

```
   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

```
1189 007400                          MOPINS  SHAP,102
1211 007400                                  LIST
1212
1213
1214
1215
1216
1217 007400                          DOPINS  RHO,102
1313 007400                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```



```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 53
```

DEFINITIONS   ERRORS

```
1598 007400                          MOPINS  IOTA,72
1632 007400                                  LIST
1633
1634
1635
1636
1637
1638 007400                          DOPINS  IND,72
1699 007400                                  LIST
```

```
APLOOK  MACRO M1113  08-FEB-80 09:26  PAGE 54
DEFINITIONS     ERRORS

  1701 007400                          DOPINS  1COM,124
  1709 007400                                  LIST
  1710
  1711
  1712
  1713
  1714
  1715 007400                          DOPINS  COMP,60,T.DIM
  1789 007400                                  LIST
```



```
APLOOK  MACRO M1113  08-FEB-80 09:26  PAGE 55
```

```
   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
APLO04  MACRO M1113  08-FEB-80 09:26  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

```
   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```

DEFINITIONS ERRORS

```
 2206 007400                            DOPINS  1ROT,146
 2214 007400                                    LIST
 2215
 2216
 2217
 2218
 2219
 2220 007400                            DOPINS  ROT,144,T.DIM
 2328 007400                                    LIST
```

```
APLO04  MACRO M1113  08-FEB-80 09:26  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

```
                                        DOPINS  TRAN,126
                                        .SBTTL  MIXED FUNCTION          DYADIC TRANSPOSE
                                        ;
                                        ;YES FOLKS, THIS IS THE MAGIC DYADIC TRANSPOSE ROUTINE.  IN THE SPECIAL
                                        ;CASE IT IS THE ''INVERSE PERMUTATION OPERATOR'', WHILE IN THE GENERAL
                                        ;CASE IT PULLS DIAGONAL PLANES (WHATEVER THEY MAY BE) OUT OF ARRAYS.
                                        ;ANYONE WHO UNDERSTANDS HOW AND WHY THIS ROUTINE WORKS, WELL ENOUGH TO
                                        ;EXPLAIN IT TO SOMEONE ELSE SHOULD IMMEDIATELY CONTACT OMSI COMPUTING
                                        ;AT THE ADDRESS ON THE FRONT PAGE AND APPLY FOR A JOB AS A SYSTEMS
                                        ;PROGRAMMER
                                        ;
                                        ;
                                        ;VARIABLE ASSIGNMENTS:
                                        ;
                                        ;  OPTMP0 - POINTER TO START OF STRIDE LIST ON SP STACK
                                        ;       1 - END OF STRIDE LIST / START OF ARG LIST
                                        ;       2 - END OF ARG LIST / START OF RESULT DIM/STRIDE LIST
                                        ;       3 - LITERAL/NUMERIC FLAG
                                        ;       4 - RIGHT ARG DIM COUNT (= LENGTH OF LEFT ARG)
                                        ;       5 - OFFSET FROM ARG LIST TO STRIDE LIST (= 2*OPTMP4)
                                        ;       6 - RIGHT ARG DIM LIST POINTER
                                        ;       7 - RESULT DIM COUNT
                                        ;
                                        ;GOOD LUCK!
                                        ;


                                        ;
                                        ;CHECK ARGUMENT CONFORMITY
                                        ;
000000                                          CALL    GETREG          ;OBTAIN POINTERS TO ARGS
000004  032700  000006                          BIT     #A.NUM,R0       ;LEFT ARG NUMERIC?
000010  001001                                  BNE     2$              ;YES

000012  104417                  1$:             DOMERR                  ;BAD ARGS

000014  116402  000002          2$:             MOVB    A.DIMS(R4),R2   ;GET RANK OF RIGHT ARG
000020                                          CONINC  R2              ;;CHEAT FOR SCALAR
000024  026302  000004                          CMP     A.ALEN(R3),R2   ;SAME AS LENGTH OF LEFT ARG?
000030  001401                                  BEQ     4$              ;YUP

000032  104411                  3$:             RANERR                  ;OOPS

000034  122763  000001  000002  4$:             CMPB    #1,A.DIMS(R3)   ;LEFT ARG A VECTOR OR SCALAR?
000042  002773                                  BLT     3$              ;MUST BE!
                                        ;
                                        ;CREATE STRIDE LIST FOR RIGHT ARG
                                        ;
000044  010637  005066'                         MOV     SP,OPTMP0       ;POINT TO START OF STRIDE LIST ON SP STACK
000050  010200                                  MOV     R2,R0           ;COPY DIM COUNT FOR RIGHT ARG
000052  010237  005076'                         MOV     R2,OPTMP4       ;THIS IS A USEFUL NUMBER
000056  006300                                  ASL     R0              ;CONVERT TO WORD COUNT
000060  062704  000006                          ADD     #A.DIM1,R4      ;POINT TO START OF RIGHT ARG'S DIM LIST
000064  060004                                  ADD     R0,R4           ;POINT PAST END OF LIST
000066  012701  000001                          MOV     #1,R1           ;INITIAL STRIDE
000072  013700  003212'                         MOV     RARGTY,R0       ;Get type of Right Arg
000076  016037  000044' 005074'                 MOV     SIZTAB(R0),OPTMP3 ;Get the shift size of arg
```

```
2474 000104  000405                          BR      7$              ;AND START UP STRIDE CALCULATOR
2475
2476 000106                          6$:     MUL     -(R4),R1        ;ADD NEXT STRIDE FACTOR
2477 000120  010103                  7$:     MOV     R1,R3           ;COPY CURRENT STRIDE
2478 000122  010346                          MOV     R3,-(SP)        ;PUSH STRIDE FOR THIS DIM
2479 000124                                  SOB     R2,6$           ;LOOP ONCE PER DIM
2480                                 ;
2481                                 ;CONVERT LEFT ARG TO INTEGERS AND PLACE ON SP STACK IN
2482                                 ;REVERSE ORDER.
2483                                 ;
2484 000130  010637  005070'                 MOV     SP,OPTMP1       ;END OF STRIDE LIST/START OF ARG LIST
2485 000134  013702  005076'                 MOV     OPTMP4,R2       ;GET LENGTH OF LEFT ARG
2486 000140  010203                          MOV     R2,R3           ;COPY COUNT
2487 000142  013701  003170'                 MOV     LARGTY,R1       ;Get type of Left Arg
2488 000146                                  ASH     SIZTAB(R1),R2   ;CONVERT TO NUMERIC COUNT
2489 000162  063702  003164'                 ADD     LARGPT,R2       ;POINT PAST END OF LEFT ARG'S VALUES
2490 000166  016137  000000G 003176'         MOV     ILARGC(R1),LARGCV ;Convert Left to Integer
2491 000174  005437  003160'                 NEG     LARGFL          ;We're Decrementing
2492 000200  063702  003160'                 ADD     LARGFL,R2       ;While we're at it, adjust pointer
2493
2494 000204                          8$:     CALL    @LARGCV         ;GET NEXT ARG
2495 000210                                  PUSH    LARGVA          ;Push onto SP Stack
2496 000214  003676                          BLE     1$              ;DO ELEMENTARY BOUNDS CHECKING
2497 000216                                  SOB     R3,8$           ;ONCE PER ARG
2498
2499                                 ;
2500                                 ;MAKE SURE LEFT ARG IS A 'DENSE INTEGER LIST'' AND CALCULATE
2501                                 ;THE RANK OF THE RESULT
2502                                 ;
2503 000222  010637  005072'                 MOV     SP,OPTMP2       ;END OF ARG LIST/START OF RESULT DIM/STRIDES
2504 000226  013703  005076'                 MOV     OPTMP4,R3       ;GET LENGTH OF ARG LIST
2505 000232  012701  000001                  MOV     #1,R1           ;MAX DIM IS INITIALLY 1
2506 000236  005000                          CLR     R0              ;NO HITS YET
2507
2508 000240  010604                  9$:     MOV     SP,R4           ;POINT TO END OF ARG LIST
2509 000242  005046                          CLR     -(SP)           ;CLEAR HIT FLAG
2510 000244  010302                          MOV     R3,R2           ;RESET COUNT
2511
2512 000246  022401                  10$:    CMP     (R4)+,R1        ;HIT?
2513 000250  001002                          BNE     11$             ;NOPE
2514 000252  005216                          INC     @SP             ;YES, MARK A HIT ON THIS PASS
2515 000254  005200                          INC     R0              ;ADD TO TOTAL HITS FOUND
2516 000256                          11$:    SOB     R2,10$          ;LOOP ONCE PER ARG
2517 000262  005726                          TST     (SP)+           ;ANY HITS ON THIS PASS?
2518 000264  001402                          BEQ     12$             ;NO, ALL DONE
2519 000266  005201                          INC     R1              ;YES, LOOK FOR NEXT DIM
2520 000270  000763                          BR      9$              ;AND MAKE ANOTHER PASS
2521
2522 000272  020003                  12$:    CMP     R0,R3           ;FIND ENOUGH HITS?
2523 000274  001246                          BNE     1$              ;NO, NOT DENSE INTEGER LIST
2524 000276  005301                          DEC     R1              ;CORRECT THE COUNT OF DIFFERENT ARGS
2525 000300  003644                          BLE     1$              ;WHAT???
2526                                 ;
2527                                 ;NOW WE KNOW HOW MANY DIMENSIONS IN THE RESULT.  IT IS STILL
2528                                 ;NECESSARY TO CALCULATE THE RESULT'S SHAPE BEFORE WE CAN
2529                                 ;ALLOCATE AN ARRAY BLOCK FOR IT.  WHILE CALCULATING THE SIZE
2530                                 ;OF THE RESULT, WE CAN ALSO CALCULATE THE STRIDES NEEDED BY
```

```
2531                                    ;'NXTIND' WHICH ACTUALLY PERFORMS THE TRANSPOSITION.
2532                                    ;
2533 000302  010137  005104'    DTRAN1: MOV     R1,OPTMP7       ;SAVE # OF DIMS IN RESULT
2534 000306  013700  005076'            MOV     OPTMP4,R0       ;GET LENGTH OF ARG LIST
2535 000312  005300                     DEC     R0              ;MINUS FUDGE FACTOR FOR AUTO-INCREMENT
2536 000314  006300                     ASL     R0              ;TO GET OFFSET FROM ARG LIST TO STRIDE LIST
2537 000316  010037  005100'            MOV     R0,OPTMP5       ;USED TO ACCUMULATE STRIDES FOR 'NXTIND'
2538 000322  013700  003204'            MOV     RARGBL,R0       ;POINT TO RIGHT ARG
2539 000326  062700  000006             ADD     #A.DIM1,R0      ;POINT TO DIM LIST
2540 000332  010037  005102'            MOV     R0,OPTMP6       ;AND SAVE PTR TO RIGHT ARG'S DIM LIST
2541
2542 000336  013704  005072'    1$:     MOV     OPTMP2,R4       ;POINT TO END OF ARG LIST
2543 000342  013703  005102'            MOV     OPTMP6,R3       ;POINT TO RIGHT ARG'S DIM LIST
2544 000346  013702  005076'            MOV     OPTMP4,R2       ;COUNT OF ARGS
2545 000352  012700  001000             MOV     #1000,R0        ;FORCE FIRST ARG TO BECOME NEW MINIMUM
2546 000356  005046                     CLR     -(SP)           ;CLEAR SPOT FOR RESULT'S STRIDE
2547
2548 000360  022401              2$:    CMP     (R4)+,R1        ;ARG MATCH CURRENT DIM?
2549 000362  001010                     BNE     3$              ;NOPE
2550 000364  063704  005100'            ADD     OPTMP5,R4       ;POINT INTO STRIDE LIST
2551 000370  061416                     ADD     @R4,@SP         ;ACCUMULATE STRIDE FOR THIS DIM
2552 000372  163704  005100'            SUB     OPTMP5,R4       ;RESTORE POINTER
2553 000376  021300                     CMP     @R3,R0          ;IS THIS DIM SMALLER THAN CURRENT DIM?
2554 000400  002001                     BGE     3$              ;NO, KEEP CURRENT VALUE FOR MIN
2555 000402  011300                     MOV     @R3,R0          ;ELSE GET NEW CANDIDATE FOR MINIMUM DIM
2556 000404  005723              3$:    TST     (R3)+           ;BUMP POINTER TO NEXT DIM
2557 000406                             SOB     R2,2$           ;LOOP ONCE PER ARG TO FIND MIN FOR THIS DIM
2558 000412  010046                     MOV     R0,-(SP)        ;WHICH BECOMES LENGTH OF DIM IN RESULT
2559 000414                             SOB     R1,1$           ;AND CYCLE FOR NEXT DIMENSION
2560                                    ;
2561                                    ;MOVE RESULT'S DIM/STRIDE LIST UP TO CLEAR MOST OF THE
2562                                    ;JUNK FROM THE SP STACK.
2563                                    ;
2564 000420  013704  005066'            MOV     OPTMP0,R4       ;POINT TO START OF STRIDE LIST
2565 000424  013703  005072'            MOV     OPTMP2,R3       ;POINT TO START OF DIM/STRIDE LIST
2566 000430  013702  005104'            MOV     OPTMP7,R2       ;GET COUNT OF DIMS IN DIM/STRIDE LIST
2567 000434  012701  000001             MOV     #1,R1           ;MULTIPLICATIVE IDENTITY
2568
2569 000440  014344              4$:    MOV     -(R3),-(R4)     ;COPY STRIDE
2570 000442  014300                     MOV     -(R3),R0        ;GET DIM
2571 000444  010044                     MOV     R0,-(R4)        ;COPY DIM
2572 000446                             MUL     R0,R1           ;ACCUMULATE TOTAL SIZE OF RESULT
2573 000460                             SOB     R2,4$           ;ONCE PER RESULT DIM
2574
2575 000464  010406                     MOV     R4,SP           ;WIPE OUT REST OF STACK
2576                                    ;
2577                                    ;CALCULATE SIZE OF RESULT, ALLOCATE ARRAY BLOCK FOR IT, AND
2578                                    ;FILL IN ITS HEADER
2579                                    ;
2580 000466                             PUSH    R1              ;SAVE ALEN OF RESULT
2581 000470                             ASH     OPTMP3,R1       ;INCREASE SIZE IF NUMERIC
2582 000504  013700  005104'            MOV     OPTMP7,R0       ;GET # OF DIMS IN RESULT
2583 000510                             PUSH    R0              ;SAVE IT
2584 000512  006300                     ASL     R0              ;CONVERT TO WORD COUNT
2585 000514  062700  000006             ADD     #A.DIM1,R0      ;TO GET HEADER SIZE
2586 000520                             PUSH    R0              ;HEADER SIZE BECOMES OFFSET TO DATA
2587 000522  060100                     ADD     R1,R0           ;TOTAL SIZE OF RESULT BLOCK NEEDED
```

```
2588 000524                                    CALL    GETFRA          ;Get result, SI frame for it
2589 000530  062620                            ADD     (SP)+,(R0)+     ;Add in Offset to Data
2590 000532  012601                            MOV     (SP)+,R1        ;GET DIM COUNT
2591 000534  110120                            MOVB    R1,(R0)+        ;SAVE # OF DIMS
2592 000536  113710  003212'                   MOVB    RARGTY,(R0)     ;Set the Type
2593 000542  001003                            BNE     5$              ;YUP
2594 000544  042765  002000  177774            BIC     #T.NUM,SI.BIT(SI);NOPE, FIX BAD ASSUMPTIONS
2595
2596 000552  152720  000200          5$:       BISB    #A.ASB,(R0)+    ;Indicate Array
2597 000556  012620                            MOV     (SP)+,(R0)+     ;FILL IN ALEN
2598 000560  010604                            MOV     SP,R4           ;COPY POINTER TO END OF DIM/STRIDE LIST
2599
2600 000562  012420                    6$:     MOV     (R4)+,(R0)+     ;COPY DIM
2601 000564  005724                            TST     (R4)+           ;SKIP OVER STRIDE
2602 000566                                    SOB     R1,6$           ;ONCE PER DIM IN RESULT
2603
2604 000572  010037  003136'                   MOV     R0,RESULT       ;SAVE POINTER TO RESULT'S DATA
2605 000576  010037  003300'                   MOV     R0,NINPT2       ;SAVE POINTER PAST DIM LIST FOR 'NXTIND'
2606 000602  005037  003302'                   CLR     NINPT1          ;THIS IS NOT REGULAR SUBSCRIPTING!!
2607 000606  013737  003212' 003306'          MOV     RARGTY,NINTYP   ;MARK AS NUMERIC OR LITERAL DATA
2608 000614  010637  003304'                   MOV     SP,NINSTK       ;SAVE PTR TO START OF INDEX FRAMES
2609                                   ;
2610                                   ;PLACE LIST OF INDEX FRAMES ON SP STACK FOR 'NXTIND' ROUTINE.
2611                                   ;
2612 000620  010604                    DOTRAN: MOV     SP,R4           ;COPY POINTER TO END OF DIM LIST
2613 000622  013702  005104'                   MOV     OPTMP7,R2       ;GET COUNT OF DIMS IN RESULT
2614
2615 000626  005046                    7$:     CLR     -(SP)           ;COUNT
2616 000630  005046                            CLR     -(SP)           ;ARRAY POINTER
2617 000632  012446                            MOV     (R4)+,-(SP)     ;LENGTH OF THIS DIMENSION
2618 000634  005046                            CLR     -(SP)           ;OLD VALUE FLAG
2619 000636  012446                            MOV     (R4)+,-(SP)     ;STRIDE FOR THIS DIM
2620 000640                                    SOB     R2,7$           ;ONCE PER DIM
2621                                   ;
2622                                   ;INITIALIZE 'NXTIND'
2623                                   ;
2624 000644  012703  000001                    MOV     #1,R3           ;START AT LOWEST LEVEL
2625 000650  010604                            MOV     SP,R4           ;COPY SP POINTER
2626 000652  062704  000012                    ADD     #I.LEN,R4       ;POINT TO REF POINT OF DEEPEST FRAME
2627 000656  013702  003206'                   MOV     RARGPT,R2       ;POINT TO FIRST ARRAY VALUE IN ARG
2628 000662                                    CALL    NXTIND          ;INITIALIZE EVERYTHING
2629 000666  102401                            BVS     8$              ;INITIALIZATION SUCCESSFUL
2630 000670  104402                            SYSERR                  ;GADS!
2631
2632 000672  013746  003136'           8$:     MOV     RESULT,-(SP)    ;SAVE POINTER TO RESULT
2633 000676  012700  000001                    MOV     #1,R0           ;BYTE COUNT OF DATA LENGTH
2634 000702                                    ASH     OPTMP3,R0       ;CONVERT IF NUMERIC
2635 000716  010037  005074'                   MOV     R0,OPTMP3       ;AND SAVE BYTE COUNT LENGTH OF DATA
2636 000722  013703  005104'                   MOV     OPTMP7,R3       ;GET COUNT OF DIMS
2637                                   ;
2638                                   ;FINALLY DO THE TRANSPOSITION!
2639                                   ;
2640 000726                            9$:     CALL    NXTIND          ;GET NEXT POINTER
2641 000732  102412                            BVS     11$             ;ALL DONE
2642 000734                                    POP     R0              ;RESTORE RESULT POINTER
2643 000736  013701  005074'                   MOV     OPTMP3,R1       ;LENGTH OF DATA TO TRANSFER
2644
```

```
2645 000742  112220                  10$:   MOVB    (R2)+,(R0)+     ;COPY DATA
2646 000744                                 SOB R1,10$
2647 000750  163702  005074'               SUB     OPTMP3,R2       ;RESTORE POINTER PROVIDED BY NXTIND
2648 000754                                 PUSH    R0              ;SAVE RESULT POINTER
2649 000756  000763                         BR      9$              ;AND MOVE NEXT VALUE
2650
2651 000760  013706  005066'          11$:   MOV     OPTMP0,SP       ;WIPE OUT ALL THE BROUHAHA ON THE STACK
2652 000764                                 RETURN                  ;AND RETURN TO CALLER
2654 000766                                 LIST
```

```
2656 000766                          DOPINS  ENCD,76
2856 000766                                  LIST
```

APL004 MACRO M1113 08-FEB-80 09:26 PAGE 62
MIXED FUNCTION DYADIC TRANSPOSE

```
2858 000766                          DOPINS  DECD,64
2988 000766                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 63
MIXED FUNCTION          DYADIC TRANSPOSE

   2990 000766                          MOPINS  GDWN,136,T.DIM
   3001 000766                                  LIST
   3002
   3003
   3004 000766                          MOPINS  GDUP,140,T.DIM
   3219 000766                                  LIST
```

```
3221 000766                          MOPINS  MMD,122
3316 000766                                  LIST
```

```
APLOO4  MACRO M1113  08-FEB-80 09:26  PAGE 65
MIXED FUNCTION          DYADIC TRANSPOSE

   3318 000766                          DOPINS  DMD,122
```

```
    3864 000766                                    LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 67
MIXED FUNCTION          DYADIC TRANSPOSE

   3866 000766                          DOPINS  MEM,70
   3921 000766                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 68
MIXED FUNCTION          DYADIC TRANSPOSE

   3923 000766                          MOPINS  NUB,106
   3986 000766                                  LIST
   3987
   3988
   3989
   3990
   3991
   3992 000766                          DOPINS  UNON,106
   4022 000766                                  LIST
   4023
   4024
   4025
   4026
   4027
   4028 000766                          DOPINS  INTR,66
   4042 000766                                  LIST
   4043
   4044
   4045
   4046
   4047
   4048 000766                          DOPINS  SDIF,50
   4161 000766                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 69
MIXED FUNCTION          DYADIC TRANSPOSE

   4163 000766                          DOPINS  SCON,234
   4179 000766                                  LIST
   4180
   4181
   4182 000766                          DOPINS  CONT,112
   4200 000766                                  LIST
   4201
   4202
   4203 000766                          DOPINS  PSUB,236
   4209 000766                                  LIST
   4210
   4211
   4212 000766                          DOPINS  SUBS,116
   4288 000766                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 70
MIXED FUNCTION          DYADIC TRANSPOSE

   4290 000766                          MOPINS  EXE,70
   4336 000766                                  LIST
```

```
                                MOPINS  ENCD,76
                                .SBTTL  MIXED FUNCTION          MONADIC ENCODE (QUOTE)

                                ;
                                ;RETURNS CHARACTER VALUE OF ARGUMENT.  IF ARGUMENT IS NUMERIC, IT IS
                                ;CONVERTED TO ITS CHARACTER REPRESENTATION.  IF THE ARGUMENT IS A
                                ;STRING, IT IS ASSUMED TO BE THE NAME OF AN IDENTIFIER.  IF THE
                                ;IDENTIFIER IS A FUNCTION, THE RESULT IS THE LISTING OF THE FUNCTION,
                                ;ELSE THE CHARACTER VALUE OF THE IDENTIFIER IS RETURNED AS ABOVE.
                                ;

000766                                  PUSH    <CHANEX,CHANEX+C.BUFF,OCHAN,CHANEX+C.CCNT,CHANEX+C.CPTR,R4>
                                                                ;SAVE OUTPUT POINTERS
001014  105037  000403'                 CLRB    TMODE           ;FORCE APL MODE
001020  005037  003274'                 CLR     ALLINT          ;NORMAL PACKING
001024                                  CALL    SETBIG          ;SET UP OUTPUT BUFFER
001030  013700  003120'                 MOV     MONBLK,R0       ;POINT TO ARG
001034  132760  000006  000003         BITB    #A.NUM,A.BITS(R0);NUMERIC?
001042  001436                         BEQ     3$              ;NOPE

001044  011604                 1$:     MOV     (SP),R4         ;Restore the operator number
001046                                  CALL    APRNT1          ;CONVERT NUMERIC ARG TO CHARACTERS

001052  113737  000402' 000403' 2$:     MOVB    OMODE,TMODE     ;RESTORE TTY MODE
001060  016501  177776                 MOV     SI.VAL(SI),R1   ;POINT TO RESULT
001064  011100                         MOV     @R1,R0          ;POINT TO RESULT DATA
001066  013702  003044'                 MOV     CHANEX+C.CPTR,R2;POINT PAST END OF DATA IN BUFFER
001072  160002                         SUB     R0,R2           ;CALCULATE LENGTH OF BUFFER
001074  010261  000004                 MOV     R2,A.ALEN(R1)   ;REAL ALEN
001100  010261  000006                 MOV     R2,A.DIM1(R1)   ;REAL DIM1
001104                                  CALL    CARVE           ;KILL UNUSED PORTION OF RESULT
001110                                  POP     <R4,CHANEX+C.CPTR,CHANEX+C.CCNT,OCHAN,CHANEX+C.BUFF,CHANEX>
                                                                ;RESTORE OUTPUT POINTERS
001136                                  RETURN
                                ;
                                ;CHECK FOR SYMBOL IN LITERAL ARG
                                ;
001140  016003  000004          3$:     MOV     A.ALEN(R0),R3   ;GET LENGTH OF ARG
001144  001742                         BEQ     2$              ;OOPS!
001146  011004                         MOV     @R0,R4          ;POINT TO DATA
001150  112402                         MOVB    (R4)+,R2        ;GET FIRST CHAR
001152  005303                         DEC     R3              ;COUNT IT
001154  042702  177400                 BIC     #UPPER,R2       ;KILL GARBAGE
001160  132762  000010  000000G        BITB    #C.IDEN,CHRTAB(R2);IDENTIFIER?
001166  001014                         BNE     5$              ;YES

001170                          4$:     POP     <R4,CHANEX+C.CPTR,CHANEX+C.CCNT,OCHAN,CHANEX+C.BUFF,CHANEX>
                                                                ;RESTORE OUTPUT POINTERS
001216  104417                 7$:     DOMERR                  ;NOT IDENTIFIER

001220                          5$:     CALL    SYMBOL          ;LOOK UP IDENTIFIER
001224  005703                         TST     R3              ;END OF ARG?
001226  002403                         BLT     6$              ;YES
001230  122702  000015                 CMPB    #CHR.CR,R2      ;CR?
001234  001355                         BNE     4$              ;NOPE
001236  011000                 6$:     MOV     @R0,R0          ;GET SYMBOL'S VALUE
001240  001704                         BEQ     2$              ;JUST A NULL!
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 71-1
MIXED FUNCTION          MONADIC ENCODE (QUOTE)

   4395 001242  005760  000002                  TST     A.DIMS(R0)          ;CHECK SOURCE
   4396 001246  002676                          BLT     1$                  ;IT IS AN ARRAY, RETURN CHAR VALUE
   4397 001250  032760  040000  000002          BIT     #S.FLB,A.DIMS(R0)   ;IS IT A FUNCTION?
   4398 001256  001744                          BEQ     4$                  ;No, DOMAIN ERROR
   4399 001260  132760  000004  000003          BITB    #F.LOC,R.BITS(R0)   ;It is a function, Is it locked?
   4400 001266  001340                          BNE     4$                  ;Yes, Then don't quote it, Domain error
   4401 001270  010037  005012'                 MOV     R0,FHEADR           ;SAVE POINTER TO FN HEADER
   4402 001274  010001                          MOV     R0,R1               ;COPY HEADER POINTER
   4403 001276  113746  003052'                 MOVB    CHANEX+C.WIDR,-(SP);SAVE OLD WIDTH
   4404 001302  105037  003052'                 CLRB    CHANEX+C.WIDR       ;CLEAR WIDTH RESET
   4405 001306  105037  003050'                 CLRB    CHANEX+C.WIDC       ;AND CURRENT WIDTH
   4406 001312                                  CALL    FNLIST              ;LIST THE FUNCTION
   4407 001316  112637  003052'                 MOVB    (SP)+,C.WIDR+CHANEX;RESTORE WIDTH
   4408 001322  113737  003052' 003050'        MOVB    CHANEX+C.WIDR,CHANEX+C.WIDC;RESET CURRENT WIDTH
   4409 001330  000650                          BR      2$                  ;AND RETURN
   4411 001332                                  LIST
```

```
                                        MOPINS  FMT,160
                                        .SBTTL  MIXED FUNCTION          MONADIC FORMAT

001332  132737  000006  003124'         BITB    #A.NUM,MONTYP   ;Is the argument character type?
001340  001004                          BNE     1$              ;No, then we have to work a little
001342  013704  003120'                 MOV     MONBLK,R4       ;Yes, Just copy the argument
001346                                  CALLR   SAME            ;  and return
001352                                  CALL    M$ENCD          ;Call Quote to convert the arg to chars
001356                                  CALLR   CMPRSS          ;Now delete the <CR><LF> pairs and return
001362                                  LIST
```

```
001362                                  DOPINS  FMT,160
                                        .SBTTL  MIXED FUNCTION          DYADIC FORMAT

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       Dyadic Thorn or Format
                                        ;
                                        ; Format converts a numeric array into a character array of the same
                                        ; shape.  The dimensions of the result are the same as the numeric array,
                                        ; except the last which in general will be larger since a number is
                                        ; represented by a vector of characters.  The right argument is the array
                                        ; to be formatted, and the left argument controls the formatting.  The
                                        ; left argument may be an integer scalar or singleton vector, a pair of
                                        ; integers, or an integer vector of length 2n, where n is the last dimension
                                        ; of the right argument.  If the left argument is a single integer then
                                        ; this integer is the precision for the entire formatted array.  If the
                                        ; left argument is a pair of integers then they are the width and precision
                                        ; respectively, for the entire formatted array. And if the left argument is
                                        ; a 2n vector then each pair is the width and precision which controls the
                                        ; formatting of the cooresponding column of the right argument.
                                        ;  The width specifies the width of the columns of the formatted array.  If
                                        ; the width is omitted or zero then the width used for the entire array or
                                        ; cooresponding column is the minimum plus one required to represent each
                                        ; number in the array or column.  I.e. the minimum required leaving one space
                                        ; between each column.  The precision may be either positive or negative.
                                        ; If it is negative then scientific notation is used and the precision gives
                                        ; the number of digits to be used in the fractional part.  If the precision
                                        ; is positive then normal decimal notation is used and the precision gives
                                        ; the number of digits to be used to the right of the decimal point.
                                        ;  If the width is not sufficient for the precision specified then a DOMAIN
                                        ; error results.  If a precision greater than the machine precision is
                                        ; specified then the missing significant digits are indicated by zeros for
                                        ; integers and by underscores for floating point numbers.  Numbers are
                                        ; rounded to the specified precision.
                                        ;
                                        ;  Written by: Peter Haynes     Date: May 1978
                                        ;
                                        ;  Use of varibles:
                                        ;
                                        ;       OPTMP0 - Pointer to format vector
                                        ;       OPTMP1 - Number of columns in the right argument
                                        ;       OPTMP2 - Length of a result row in bytes
```

```
                                        ;       OPTMP3 - Number of rows in the result
                                        ;       OPTMP4 - Exponent for Scientific notation
                                        ;       OPTMP5 - Sign flag for the right argument
                                        ;       OPTMP6 - Conversion routine to use, NORM or SCI
                                        ;       OPTMP7 - Result increment constant
                                        ;       OPTMP8 - Scratch location
                                        ;       OPTMP9 - Pad character, zero or underscore
                                        ;       OPTM10 - Maximum number of digits of precision for RTS
                                        ;       RESULT - Used to hold negative sign character for current terminal
                                        ;
                                        ;       AC0    - Scratch
                                        ;       AC1    - Fill routine conversion constant, 10.0
                                        ;       AC2    - Right argument to be converted
                                        ;       AC3    - Scratch
                                        ;
                                        ; Subroutines:
                                        ;
                                        ;       SCANCL - Used to find the minimum width to represent a column
                                        ;       NORM   - Converts numbers to normal notation
                                        ;       SCI    - Converts numbers to scientific notation
                                        ;       FILL   - Fills in the result field with characters
                                        ;       LOG    - Computes logrithms base 10
                                        ;--
001362  132737  000006  003170'          BITB    #A.NUM,LARGTY   ;Left argument a character?
001370  001404                          BEQ     1$              ;Yes,  Domain error
001372  132737  000006  003212'          BITB    #A.NUM,RARGTY   ;Right argument a character?
001400  001001                          BNE     2$              ;No, so ok
001402  104417                  1$:     DOMERR                  ;DOMAIN ERROR
001404  013703  003162'          2$:     MOV     LARGBL,R3       ;Get the left argument pointer
001410  122763  000001  000002          CMPB    #1,A.DIMS(R3)   ;Left argument a vector or scalar?
001416  103001                          BHIS    4$              ;Yes
001420  104411                  3$:     RANERR                  ;No, RANK ERROR
001422  005737  003166'          4$:     TST     LARGLN          ;Left argument a null vector?
001426  001774                          BEQ     3$              ;Yes, so error
001430  013703  003210'                  MOV     RARGLN,R3       ;Right argument a null array?
001434  001771                          BEQ     3$              ;Yes, so error
001436  013704  003204'                  MOV     RARGBL,R4       ;Get the right argument pointer
001442  116400  000002                  MOVB    A.DIMS(R4),R0   ;Get the number of dimensions in right argument
001446                                  CONINC  R0              ;Fudge for scalars
001452  006300                          ASL     R0              ;Make it into word offset
001454  060400                          ADD     R4,R0           ;Add in reference pointer
001456  062700  000004                  ADD     #A.ALEN,R0      ;Overhead too
001462  011000                          MOV     (R0),R0         ;Get the last dimension
001464  010037  005070'                  MOV     R0,OPTMP1       ;Save it
001470  010001                          MOV     R0,R1           ;Copy it
001472                                  ASH     RARGSF,R1       ;Get the right argument row length in bytes
001506  010137  003202'                  MOV     R1,RARGFL       ;Save it
001512  005002                          CLR     R2              ;Divide setup
001514                                  DIV     R0,R2           ;Divide by the number of columns
001526  010237  005074'                  MOV     R2,OPTMP3       ;Save the number of rows
001532  013701  003212'                  MOV     RARGTY,R1       ;Get the right argument type
001536  016137  000000G 003220'          MOV     FRARGC(R1),RARGCV  ;Setup argument fetch routine, to float
001544  013701  003170'                  MOV     LARGTY,R1          ;Get left argument size
001550  016137  000000G 003176'          MOV     ILARGC(R1),LARGCV  ;Setup left arg fetch routine, to integer
001556  006300                          ASL     R0              ;Double number of columns for words
001560  013702  003166'                  MOV     LARGLN,R2       ;Get left argument length
001564  022702  000001                  CMP     #1,R2           ;Is it one?
```

```
4528 001570 001406                          BEQ     5$              ;Yes, ok
4529 001572 022702 000002                   CMP     #2,R2           ;Is it two?
4530 001576 001403                          BEQ     5$              ;Yes, ok
4531 001600 020002                          CMP     R0,R2           ;No, Is it = two times the number of columns?
4532 001602 001401                          BEQ     5$              ;Yes, ok
4533 001604 104412                          LENERR                  ;No, LENGTH ERROR
4534 001606 062700 000004           5$:     ADD     #4,R0           ;Add in overhead
4535 001612                                 CALL    GETCOR          ;Get space for the format vector
4536 001616                                 SIFRAM                  ;Build an SI frame
4537 001630 005720                          TST     (R0)+           ;Move up one
4538 001632 012720 100000                   MOV     #S.ASB,(R0)+    ;Make it an array block
4539 001636 010037 005066'                  MOV     R0,OPTMP0       ;Save pointer to format vector
4540 001642 013701 005070'                  MOV     OPTMP1,R1       ;Get the number of columns
4541 001646 013702 003164'                  MOV     LARGPT,R2       ;Get pointer to left argument
4542 001652 013703 003206'                  MOV     RARGPT,R3       ;Get pointer to right argument
4543 001656 005037 005072'                  CLR     OPTMP2          ;For length of row in result
4544 001662 005004                          CLR     R4              ;Assume there is no width
4545 001664                                 LDF     #^F10.0,AC1     ;Setup the scaling factor
4546 001674 022737 000001 003166'           CMP     #1,LARGLN       ;Just the precision?
4547 001702 001407                          BEQ     7$              ;Yes
4548
4549                                ;+
4550                                ;  Find the number of bytes in the result row, i.e. the size of the last dim
4551                                ;-
4552
4553 001704                          6$:     CALL    @LARGCV         ;Get the width
4554 001710 013704 003200'                  MOV     LARGVA,R4       ;Save it in a register
4555 001714 022704 000377                   CMP     #255.,R4        ;Is the width reasonable?
4556 001720 103630                          BLO     1$              ;No, DOMAIN Error
4557 001722                         7$:     CALL    @LARGCV         ;Get the precision
4558 001726 023727 003200' 000177           CMP     LARGVA,#127.    ;Is the precision reasonable?
4559 001734 003222                          BGT     1$              ;No, DOMAIN Error
4560 001736 023727 003200' 177601           CMP     LARGVA,#-127.   ;Very reasonable?
4561 001744 002616                          BLT     1$              ;No, DOMAIN Error
4562 001746                         8$:     PUSH    R4              ;Save the width argument
4563 001750 001002                          BNE     9$              ;Is the width = 0 ?
4564 001752                                 CALL    SCANCL          ;Yes, Have to compute width, Blaah!!!!
4565 001756 060437 005072'          9$:     ADD     R4,OPTMP2       ;Accumulate the row width
4566 001762 110420                          MOVB    R4,(R0)+        ;Store width in low byte of vector
4567 001764 113720 003200'                  MOVB    LARGVA,(R0)+    ;Pack in the precision
4568 001770 063703 003216'                  ADD     RARGLE,R3       ;Point to next right argument column
4569 001774                                 POP     R4              ;Restore the width argument
4570 001776 005301                          DEC     R1              ;Count the column
4571 002000 001405                          BEQ     10$             ;Last column?
4572 002002 022737 000002 003166'           CMP     #2,LARGLN       ;No, Only one pair?
4573 002010 002735                          BLT     6$              ;No, Loop on next column
4574 002012 000755                          BR      8$              ;Yes, Use it again
4575
4576 002014 013700 005072'          10$:    MOV     OPTMP2,R0       ;Get the row length
4577 002020                                 MUL     OPTMP3,R0       ;Times the number of rows
4578 002034 103001                          BCC     11$             ;Too big?
4579 002036 104403                          WSFULL                  ;Yes, so WORKSPACE is FULL
4580 002040 010137 005104'          11$:    MOV     R1,OPTMP7       ;Save the result increment constant
4581 002044 012700 000006                   MOV     #A.DIM1,R0      ;Get the overhead
4582 002050 013702 003204'                  MOV     RARGBL,R2       ;Get pointer to right argument block
4583 002054 116203 000002                   MOVB    A.DIMS(R2),R3   ;Get the number of dimensions
4584 002060 010304                          MOV     R3,R4           ;Copy it
```

```
4585 002062                                        CONINC  R3              ;Cheat for a scalar right argument
4586 002066  006303                                ASL     R3              ;Make it words
4587 002070  060300                                ADD     R3,R0           ;Add it in
4588 002072  010003                                MOV     R0,R3           ;Copy the data offset
4589 002074  060100                                ADD     R1,R0           ;Add in data space required
4590 002076                                        CALL    GETCOR          ;Get space for result
4591 002102                                        SIFRAM  ,#T.VAL         ;Build an SI frame
4592 002114  010010                                MOV     R0,(R0)         ;Fill in array block header
4593 002116  060320                                ADD     R3,(R0)+        ;  Pointer to data
4594 002120  110420                                MOVB    R4,(R0)+        ;  Dimensions
4595 002122  001006                                BNE     12$             ;  Is the Right argument a scalar?
4596 002124  122777  000001  005066'                CMPB    #1,@OPTMP0      ;  Yes, Is the Width one?
4597 002132  001402                                BEQ     12$             ;  Yes, really is a Scalar!
4598 002134  105260  177777                        INCB    -1(R0)          ;  Make result a vector
4599 002140  112720  000200            12$:        MOVB    #<A.ASB+.CHARACTER>,(R0)+
4600                                                                        ;  Mark it as an array block
4601 002144  010120                                MOV     R1,(R0)+        ;  Total length, ALEN
4602 002146  022704  000001                        CMP     #1,R4           ;  Is the right argument a scalar or vector?
4603 002152  002006                                BGE     14$             ;  Yes, Skip copying dimensions
4604 002154  005304                                DEC     R4              ;  Count of dimensions minus one
4605 002156  062702  000006                        ADD     #A.DIM1,R2      ;  Point to right argument dimensions
4606 002162  012220                    13$:        MOV     (R2)+,(R0)+     ;  Copy all but last dimension
4607 002164                                        SOB     R4,13$          ;  Done yet?
4608 002170  013720  005072'            14$:        MOV     OPTMP2,(R0)+    ;  Number of columns
4609 002174  013703  005074'                        MOV     OPTMP3,R3       ;Get the number of rows
4610 002200                                        MUL     RARGFL,R3       ;Multiply by size of right argument row
4611 002214  160337  003216'                        SUB     R3,RARGLE       ;Right argument increment constant
4612 002220  112737  000100  003136'               MOVB    #CHRNEG,RESULT  ;Assume terminal is APL type
4613 002226  112737  000106  005110'               MOVB    #CHR.UL,OPTMP9  ;For underscore too
4614 002234  105737  000402'                        TSTB    OMODE           ;Is it really?
4615 002240  001406                                BEQ     15$             ;Yes
4616 002242  112737  000137  003136'               MOVB    #CHRMNS,RESULT  ;No, Fake negative sign for TTY terminal
4617 002250  112737  000133  005110'               MOVB    #CHRBKR,OPTMP9  ;Underscore too
4618 002256  013703  003204'            15$:        MOV     RARGBL,R3       ;Point to right argument
4619 002262  116304  000003                        MOVB    A.BITS(R3),R4   ;Get the bits
4620 002266  142704  177771                        BICB    #^C<A.NUM>,R4   ;Mask out garbage
4621 002272  122704  000006                        CMPB    #.FLOAT,R4      ;Is the argument floating point type?
4622 002276  001403                                BEQ     16$             ;Yes, so pad character is correct
4623 002300  112737  000060  005110'               MOVB    #CHR..0,OPTMP9  ;No, use zero as pad for integers
4624 002306  011303                    16$:        MOV     (R3),R3         ;Get pointer to right argument data
4625 002310  012737  000006  005112'               MOV     #6,OPTM10       ;Compute maximum precision for this RTC
4626 002316  022737  000004  000062'               CMP     #4,NUMLEN       ;Is it single precision?
4627 002324  002003                                BGE     17$             ;Yes
4628 002326  062737  000012  005112'               ADD     #10.,OPTM10     ;No, it must be double precision then
4629                                    ;+
4630                                    ; Column Loop
4631                                    ;-
4632
4633 002334  012737  002706' 005102'  17$:        MOV     #NORM,OPTMP6    ;Assume normal format
4634 002342  013704  005066'                        MOV     OPTMP0,R4       ;Get pointer to format vector
4635 002346  005001                                CLR     R1
4636 002350  152401                                BISB    (R4)+,R1        ;Get the width
4637 002352  112402                                MOVB    (R4)+,R2        ;Get the precision
4638 002354  002003                                BGE     18$             ;Is the precision negative?
4639 002356  012737  003746' 005102'               MOV     #SCI,OPTMP6     ;Yes, Use scientific notation
4640 002364  010437  005066'            18$:        MOV     R4,OPTMP0       ;Save pointer to Format vector
4641 002370  013704  005074'                        MOV     OPTMP3,R4       ;Get the number of rows to process
```

```
4642
4643
4644                                    ;+
4645                                    ; Row loop
4646                                    ;-
4647
4648 002374                             19$:    CALL    @RARGCV         ;Get the next right argument
4649 002400  005037  005100'                    CLR     OPTMP5          ;Assume the right argument is positive
4650 002404                                     TSTF    AC2             ;Is it?
4651 002412                                     CFCC
4652 002416  100005                             BPL     20$             ;Yes
4653 002420  005237  005100'                    INC     OPTMP5          ;No, Mark the sign flag
4654 002424                                     NEGF    AC2             ;And make it positive
4655 002432                             20$:    CALL    @OPTMP6         ;Format it
4656 002436  063700  005072'                    ADD     OPTMP2,R0       ;Point to next row of the result
4657 002442  160100                             SUB     R1,R0           ;To the beginning
4658 002444                                     SOB     R4,19$          ;Any more rows?
4659
4660 002450  063703  003216'                    ADD     RARGLE,R3       ;No, Point to next column of right argument
4661 002454  060100                             ADD     R1,R0           ;Point to next column of the result
4662 002456  163700  005104'                    SUB     OPTMP7,R0       ;Back to first row
4663 002462  005337  005070'                    DEC     OPTMP1          ;Count this column
4664 002466  003322                             BGT     17$             ;Last column?
4665
4666
4667                                    ;+
4668                                    ; Now Kill the Temporary vector
4669                                    ;-
4670
4671 002470                                     PUSH    <-(SI),-(SI),-(SI)>  ;Save the Result SI frame
4672 002476                                     CALL    KILL                 ;Delete the temporary vector
4673 002502                                     POP     <(SI)+,(SI)+,(SI)+>  ;Restore the Result SI frame
4674 002510                                     RETURN                       ;Yes, We're done at last
4675                                    ;+
4676                                    ; This routine returns the maximun width for the current column in R4
4677                                    ;-
4678
4679 002512                             SCANCL: PUSH    <R0,R1,R2,R3>   ;Save some registers
4680 002522  013701  005074'                    MOV     OPTMP3,R1       ;Get the number of rows
4681 002526  005037  005100'                    CLR     OPTMP5          ;Clear temporary
4682 002532  012704  000001                     MOV     #1,R4           ;Set width to 1, the minimum
4683 002536  013702  003200'                    MOV     LARGVA,R2       ;Get the precision
4684 002542  002450                             BLT     7$              ;Precision is negative, Scientific notation
4685 002544  001401                             BEQ     1$              ;Precicision is zero
4686 002546  005204                             INC     R4              ;For the decimal point
4687 002550  060204                     1$:     ADD     R2,R4           ;Precision is positive, width=width+precision
4688
4689 002552                             2$:     CALL    @RARGCV         ;Get the right argument
4690 002556  005000                             CLR     R0              ;Assume the argument is positive
4691 002560                                     TSTF    AC2             ;Is it?
4692 002566                                     CFCC
4693 002572  100004                             BPL     3$              ;Yes
4694 002574  005200                             INC     R0              ;No, Add in a space for the sign
4695 002576                                     NEGF    AC2             ;And make the argument positive for the Log
4696 002604  005702                     3$:     TST     R2              ;Is the precision positive?
4697 002606  100410                             BMI     5$              ;No, this is easy
4698 002610                                     CALL    LOG             ;Yes, LOGs Blaah !!
```

```
4699 002614 005737 005076'          TST     OPTMP4          ;Is log negative?
4700 002620 100402                  BMI     4$              ;Yes, so skip adding it in
4701 002622 063700 005076'          ADD     OPTMP4,R0       ;Add in the log
4702 002626 005200              4$: INC     R0              ;And a fudge
4703 002630 020037 005100'      5$: CMP     R0,C  MP5       ;Get Max of width and temporary
4704 002634 003402                  BLE     6$              ;Temporary is larger
4705 002636 010037 005100'          MOV     R0,OPTMP5       ;Width is larger, Make it new Temporary
4706 002642                     6$: SOB     R1,2$           ;All the rows checked yet?
4707
4708 002646 063704 005100'          ADD     OPTMP5,R4       ;Yes, width=width+temporary
4709 002652                         POP     <R3,R2,R1,R0>   ;Restore the registers
4710 002662                         RETURN                  ;Done at last!!
4711
4712
4713 002664 160204              7$: SUB     R2,R4           ;Width=width+ABS(precision), Scientific form
4714 002666 062704 000004           ADD     #4,R4           ;Width=width+4, for exponent
4715 002672 022702 177777           CMP     #-1,R2          ;Is precision equal to mimus one
4716 002676 001725                  BEQ     2$              ;Yes
4717 002700 005204                  INC     R4              ;No, For the decimal point
4718 002702 000723                  BR      2$              ;Go to check column for sign
```

```
4720                                    ;+
4721                                    ;  This routine converts the number in AC2 into normal decimal notation
4722                                    ;-
4723
4724 002704  104417                    FMTDOM: DOMERR                  ;Domain error
4725 002706  023727  000062' 000004    NORM:   CMP     NUMLEN,#4       ;What is the current precision?
4726 002714  001010                            BNE     1$              ;Double, Ignore the Force
4727 002716                                    SETD                    ;Force double precision mode
4728 002722                                    LDCFD   AC2,AC2         ;Convert the Right argument to double precision
4729 002730                                    LDCFD   AC1,AC1         ;Convert the scaling factor
4730 002736                            1$:     PUSH    <R1,R2,R3,R4>   ;Save some registers
4731 002746                                    PUSH    OPTM10          ;Save the maximum precision too
4732 002752  005702                            TST     R2              ;What is the precision?
4733 002754  001405                            BEQ     3$              ;Precision equal zero?
4734 002756  100003                            BPL     2$              ;Are we doing scientific notation?
4735 002760  005202                            INC     R2              ;Yes, Adjust the precision
4736 002762  005402                            NEG     R2              ;Make it positive, P = -(P+1) = (-P)-1
4737 002764  001401                            BEQ     3$              ;Is the new precision zero?
4738
4739 002766  005301                    2$:     DEC     R1              ;No, Width = Width - 1, Account for the point
4740 002770                            3$:     PUSH    OPTMP4          ;Save previous log value for SCI routine
4741 002774                                    CALL    LOG             ;Get log of right argument
4742 003000  013704  005076'                   MOV     OPTMP4,R4       ;T2 = log (right argument)
4743 003004                                    POP     OPTMP4          ;Restore previous log value
4744 003010  005204                            INC     R4              ;Fudge the log to be size of integer part
4745 003012                                    CALL    FMTRND          ;Round the argument
4746 003016  010237  005106'                   MOV     R2,OPTMP8       ;Copy Precision
4747 003022  060402                            ADD     R4,R2           ;T1 = T2 + Precision, Digits in entire number
4748 003024  005704                            TST     R4              ;Is there an integer part, (T2>0) ?
4749 003026  003002                            BGT     4$              ;Yes
4750 003030  005301                            DEC     R1              ;No, Width = Width - 1, Account for leading 0
4751 003032  000401                            BR      5$
4752
4753 003034  160401                    4$:     SUB     R4,R1           ;Width - T2, Account for integer part
4754 003036  163701  005106'           5$:     SUB     OPTMP8,R1       ;Number of blanks = Width - T2 - Precision
4755 003042  002720                            BLT     FMTDOM          ;Number of blanks needed < 0, Domain error
4756 003044  003003                            BGT     6$              ;At least one blank needed, Width Ok
4757 003046  005737  005100'                   TST     OPTMP5          ;Zero blanks needed, Is the sign minus?
4758 003052  001314                            BNE     FMTDOM          ;Yes, Width too small, Domain error
4759 003054  005046                    6$:     CLR     -(SP)           ;Setup the zero flag
4760 003056  112703  000040                    MOVB    #CHRSPA,R3      ;Fill in leading blanks
4761 003062                                    CALL    FILL
4762 003066  010016                            MOV     R0,(SP)         ;Save sign location, i.e. zero flag
4763 003070  005737  005100'                   TST     OPTMP5          ;Is the argument negative?
4764 003074  001403                            BEQ     7$              ;No
4765 003076  113760  003136' 177777           MOVB    RESULT,-1(R0)   ;Yes, Put in a minus sign
4766 003104  005704                    7$:     TST     R4              ;Is there an integer part, (T2>0) ?
4767 003106  003003                            BGT     8$              ;Yes
4768 003110  112720  000060                    MOVB    #CHR..0,(R0)+   ;No, Fill in the leading zero
4769 003114  000431                            BR      11$             ;And go to do the fraction part
4770
4771 003116  012703  177777            8$:     MOV     #-1,R3          ;Prepare to fill in the integer part
4772 003122  023704  005112'                   CMP     OPTM10,R4       ;Requested precision > machine precision?
4773 003126  002406                            BLT     9$              ;Yes
4774 003130  010401                            MOV     R4,R1           ;No, Use T2 number of digits in integer part
4775 003132                                    CALL    FILL            ;Fill in integer part
4776 003136  160437  005112'                   SUB     R4,OPTM10       ;Account for digits used so far of max precision
```

```
4777 003142  000415                        BR      10$
4778
4779 003144  013701  005112'        9$:    MOV     OPTM10,R1       ;Use all the significant digits
4780 003150                                CALL    FILL            ;Fill in some of the integer part
4781 003154  113703  005110'               MOVB    OPTMP9,R3       ;Get the pad character
4782 003160  010401                        MOV     R4,R1           ;Pad the rest
4783 003162  163701  005112'               SUB     OPTM10,R1       ;With  T2 - DIGITS characters
4784 003166  005037  005112'               CLR     OPTM10          ;So will pad after decimal point too
4785 003172                                CALL    FILL            ;Fill in the rest of the integer part
4786
4787 003176  160402                10$:    SUB     R4,R2           ;Count the chars used so far, T1 = T1 - T2
4788 003200  005737  005106'        11$:    TST     OPTMP8          ;Is there a fractional part, (P>0) ?
4789 003204  001442                        BEQ     16$             ;No, So skip filling in fractional part
4790 003206  112720  000056               MOVB    #CHRDOT,(R0)+   ;Yes, Put in a decimal point
4791 003212  005704                        TST     R4              ;Are there any significant zeros, (T2<=0) ?
4792 003214  003013                        BGT     13$             ;No
4793 003216  010401                        MOV     R4,R1           ;Yes, Get how many
4794 003220  005401                        NEG     R1              ;Make it positive
4795 003222  023701  005106'               CMP     OPTMP8,R1       ;Is requested precision < significant zeros?
4796 003226  002002                        BGE     12$             ;Yes
4797 003230  013701  005106'               MOV     OPTMP8,R1       ;No, Use requested precision number of zeros
4798 003234  112703  000060        12$:    MOVB    #CHR..0,R3      ;Make zero the fill character
4799 003240                                CALL    FILL            ;And fill in the zeros
4800
4801 003244  012703  177777        13$:    MOV     #-1,R3          ;Prepare to fill in fractional part
4802 003250  023702  005112'               CMP     OPTM10,R2       ;Is requested precision > machine precision?
4803 003254  002012                        BGE     14$             ;No
4804 003256  013701  005112'               MOV     OPTM10,R1       ;Yes, Fill in what we can
4805 003262                                CALL    FILL            ;Fill in remaining part of fraction
4806 003266  010201                        MOV     R2,R1           ;Now we have to pad the rest of the field
4807 003270  163701  005112'               SUB     OPTM10,R1       ;With T1 - DIGITS characters
4808 003274  113703  005110'               MOVB    OPTMP9,R3       ;Get the pad character
4809 003300  000402                        BR      15$             ;And go do it
4810
4811 003302  010201                14$:    MOV     R2,R1           ;Fill in entire fractional part
4812 003304  002402                        BLT     16$             ;If there is any?
4813 003306                        15$:    CALL    FILL            ;Yes, Fill in the rest of the field
4814
4815 003312                        16$:    POP     R1              ;Get the zero flag
4816 003314  001406                        BEQ     17$             ;Only zeros printed?
4817 003316  005301                        DEC     R1              ;Yes, Point to sign
4818 003320  123711  003136'               CMPB    RESULT,(R1)     ;Is there a minus sign there?
4819 003324  001002                        BNE     17$             ;No, Don't do anything then
4820 003326  112711  000040               MOVB    #CHRSPA,(R1)    ;Yes, Change minus zero to plain zero
4821 003332                        17$:    POP     OPTM10          ;Restore the maximum precision
4822 003336                                POP     <R4,R3,R2,R1>   ;Restore the registers
4823 003346  023727  000062' 000004        CMP     NUMLEN,#4       ;Are we a single precision system?
4824 003354  001005                        BNE     18$             ;No
4825 003356                                SETF                    ;Yes, Restore floating point precision mode
4826 003362                                LDCDF   AC1,AC1         ;Restore the scaling factor too
4827 003370                        18$:    RETURN
4828                                       ;+
4829                                       ;  This routine fills in the result with characters
4830                                       ;       R0 - Pointer within result
4831                                       ;       R1 - Number of characters to copy
4832                                       ;       R3 - Pad character
4833                                       ;-
```

```
                                FILL:   TST     R1              ;Count of characters zero?
003372  005701                          BEQ     3$              ;Yes, then just return
003374  001407                          BPL     1$              ;Count positive?
003376  100001                          SYSERR                  ;No, SYSTEM Error
003400  104402                  1$:     TST     R3              ;Is this a fill character?
003402  005703                          BMI     4$              ;No, so use right argument for source
003404  100404                  2$:     MOVB    R3,(R0)+        ;Yes, use it for fill
003406  110320                          SOB     R1,2$           ;Count it, done yet?
003410                          3$:     RETURN                  ;Yes, so quit
003414

003416                          4$:     MODF    AC1,AC2         ;Multiply by 10
003424                                  STCFI   AC3,R3          ; and get integer part
003432  001402                          BEQ     5$              ;Is it zero?
003434  005066  000002                  CLR     2(SP)           ;No, Clear the zero flag
003440  062703  000060          5$:     ADD     #CHR..0,R3      ;Convert to a character
003444  110320                          MOVB    R3,(R0)+        ;Put it into the result field
003446                                  SOB     R1,4$           ;Done yet?
003452  000760                          BR      3$              ;Yes


                                ;+
                                ;  This routine computes the round constant for each argument and then
                                ;  rounds it.  A correction is made in case the magnitude of the number
                                ;  is increased.
                                ;-

003454  010203                  FMTRND: MOV     R2,R3           ;Copy the precision
003456  005704                          TST     R4              ;Is the log of the argument > 0?
003460  003401                          BLE     1$              ;No
003462  060403                          ADD     R4,R3           ;Yes, Add it to the precision
003464  020337  005112'          1$:     CMP     R3,OPTM10       ;Is the result > max digits?
003470  003402                          BLE     2$              ;No
003472  013703  005112'                  MOV     OPTM10,R3       ;Yes, Use the max digits
003476                          2$:     LDF     #^F0.5,AC0      ;Get the initial rounding constant
003506  005703                          TST     R3              ;Is the count zero?
003510  001405                          BEQ     4$              ;Yes, No adjustment needed
003512                          3$:     DIVF    AC1,AC0         ;Reduce rounding constant by 10
003520                                  SOB     R3,3$           ;Real quick
003524                          4$:     ADDF    AC0,AC2         ;Round the agrument
003532                                  CMPF    #^F1.0,AC2      ;Did the magnitude increase?
003542                                  CFCC
003546  003004                          BGT     5$              ;No
003550                                  DIVF    AC1,AC2         ;Rescale the argument
003556  005204                          INC     R4              ;Fixup the log value too
003560                          5$:     RETURN
                                ;+
                                ;  This routine computes the Log base 10 of the number in AC2, putting the
                                ;  result in OPTMP4, and leaving the number in the form: 0.xxxxxxx
                                ;  AC2 is assumed to be positive.  AC1 contains the constant 10.0,
                                ;  and AC3 is used to hold the reduction constant, 0.1 or 10.0
                                ;-

003562                          LOG:    PUSH    <R0,R1,R2>      ;Save some registers
003570  005001                          CLR     R1              ;The base 10 exponent will go here
003572                                  TSTF    AC2             ;Is the right argument positive?
003600                                  CFCC
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 73-3
MIXED FUNCTION          DYADIC FORMAT

 4891 003604  001452                        BEQ     6$              ;It is zero, we're all done!
 4892 003606                                LDF     #^F1.0,AC3      ;Get a floating one
 4893 003616                                DIVF    AC1,AC3         ;Make it 0.1, the reduction const for a plus exp
 4894 003624  012702  000001                MOV     #1,R2           ;Get the increment for a plus exponent
 4895 003630                                STEXP   AC2,R0          ;Get the exponent sign
 4896 003636  100004                        BPL     1$              ;Is it positive?
 4897 003640                                LDF     AC1,AC3         ;Get the reduction constant for a minus exponent
 4898 003646  005402                        NEG     R2              ;Get the increment for a minus exponent, -1
 4899
 4900 003650                          1$:   CMPF    #^F1.0,AC2      ;Compare the argument with 1.0
 4901 003660  005700                        TST     R0              ;Is the exponent minus?
 4902 003662  100404                        BMI     2$              ;Yes
 4903 003664                                CFCC                    ;Plus exponent, Is the argument .GE. 1.0?
 4904 003670  003404                        BLE     3$              ;Yes, so go reduce it
 4905 003672  000410                        BR      4$              ;No, So we are done
 4906 003674                          2$:   CFCC                    ;Minus exponent, Is the argument < 1.0?
 4907 003700  003405                        BLE     4$              ;No, So we are done
 4908 003702                          3$:   MULF    AC3,AC2         ;Reduce the argument by ten
 4909 003710  060201                        ADD     R2,R1           ;Count this digit
 4910 003712  000756                        BR      1$              ;Loop until we get to the range (1.0, 0.1]
 4911
 4912 003714  005700                  4$:   TST     R0              ;Was the exponent negative?
 4913 003716  100004                        BPL     5$              ;Yes
 4914 003720                                DIVF    AC1,AC2         ;Correction for negative exponents
 4915 003726  000401                        BR      6$
 4916 003730  005301                  5$:   DEC     R1              ;Correction plus exponents
 4917 003732  010137  005076'        6$:   MOV     R1,OPTMP4       ;Save the Log
 4918 003736                                POP     <R2,R1,R0>      ;Restore the registers for NORM
 4919 003744                                RETURN
 4920                                ;+
 4921                                ;  This routine formats a number into Scientific notation.  It uses NORM
 4922                                ;  to format the fractional part and then fills in the 'E' part.
 4923                                ;-
 4924
 4925 003746                          SCI:  CALL    LOG             ;Reduce the right arg to 0.xxxxEyy form
 4926 003752                                PUSH    <R1,R2,R3>      ;Save some registers
 4927 003760  162701  000004                SUB     #4,R1           ;Width = Width - 4, Account for the exponent
 4928 003764                                MULF    AC1,AC2         ;Put right argument in x.xxxEyy form
 4929 003772                                CALL    NORM            ;Use NORM to convert rest of right argument
 4930 003776  112720  000145                MOVB    #CHR..E,(R0)+   ;Fill in the character 'E'
 4931 004002  005002                        CLR     R2              ;For division
 4932 004004  005001                        CLR     R1              ;Sign flag
 4933 004006  013703  005076'              MOV     OPTMP4,R3       ;Get the exponent
 4934 004012  100004                        BPL     1$              ;Is it negative?
 4935 004014  113720  003136'              MOVB    RESULT,(R0)+    ;Yes, Fill in a minus sign
 4936 004020  005201                        INC     R1              ;Set the sign flag
 4937 004022  005403                        NEG     R3              ;Make the exponent positive
 4938 004024                          1$:   DIV     #10.,R2         ;Divide by ten
 4939 004040  001403                        BEQ     2$              ;Is it less than !10! ?
 4940 004042  062702  000060                ADD     #CHR..0,R2      ;No, Make the tens digit a character
 4941 004046  110220                        MOVB    R2,(R0)+        ;Fill it into the result
 4942 004050  062703  000060          2$:   ADD     #CHR..0,R3      ;Make the units digits into a character
 4943 004054  110320                        MOVB    R3,(R0)+        ;Fill it into the result
 4944 004056  005702                        TST     R2              ;Was there a tens digit?
 4945 004060  001002                        BNE     3$              ;Yes
 4946 004062  112720  000040                MOVB    #CHRSPA,(R0)+   ;No, Fill in a blank
 4947 004066  005701                  3$:   TST     R1              ;Was the exponent negative?
```

```
 4948 004070  001002                        BNE     4$                ;Yes
 4949 004072  112720  000040                MOVB    #CHRSPA,(R0)+     ;Pad out the rest with a blank
 4950 004076                          4$:   POP     <R3,R2,R1>        ;Restore the registers
 4951 004104                                RETURN                    ;All done
 4953 004106                                LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 74
MIXED FUNCTION          DYADIC FORMAT

   4955 004106                          MOPINS  IBM,132,,,MONIBM
   4958 004106                                  LIST
   4959
   4960
   4961 004106                          DOPINS  IBM,132
   5046 004106                                  LIST
```

```
5048 004106                          MOPINS  QUAD,74,T.NIL+T.SPOP
5075 004106                                  LIST
5076
5077
5078
5079
5080
5081 004106                          DOPINS  QUAD,74,T.NIL+T.SPOP
5097 004106                                  LIST
5098
5099
5100
5101
5102
5103 004106                          MOPINS  QQAD,142,T.NIL+T.SPOP
5135 004106                                  LIST
5136
5137
5138
5139
5140
5141 004106                          DOPINS  QQAD,142,T.NIL+T.SPOP
5152 004106                                  LIST
5153
5154
5155
5156
5157
5158 004106                          MOPINS  DQAD,134,T.NIL+T.SPOP
5192 004106                                  LIST
5193
5194
5195
5196 004106                          DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 004106                                  LIST
```

```
5201 004106                          MOPINS  BXAV,162,T.NIL
5219 004106                                  LIST
5220
5221
5222
5223
5224
5225 004106                          MOPINS  BXCT,164,T.NIL+T.SPOP
5234 004106                                  LIST
5235
5236
5237 004106                          DOPINS  BXCT,164,T.NIL+T.SPOP
5272 004106                                  LIST
5273
5274
5275
5276
5277
5278 004106                          MOPINS  BXIO,166,T.NIL+T.SPOP
5287 004106                                  LIST
5288
5289
5290 004106                          DOPINS  BXIO,166,T.NIL+T.SPOP
5302 004106                                  LIST
5303
5304
5305
5306
5307 004106                          MOPINS  BXLC,212,T.NIL
5340 004106                                  LIST
5341
5342
5343
5344
5345
5346
5347 004106                          MOPINS  BXPP,170,T.NIL+T.SPOP
5356 004106                                  LIST
5357
5358
5359 004106                          DOPINS  BXPP,170,T.NIL+T.SPOP
5381 004106                                  LIST
5382
5383
5384
5385
5386
5387
5388 004106                          MOPINS  BXPW,172,T.NIL+T.SPOP
5397 004106                                  LIST
5398
5399
5400 004106                          DOPINS  BXPW,172,T.NIL+T.SPOP
5417 004106                                  LIST
5418
5419
5420
```

```
 5421
 5422
 5423 004106                        MOPINS  BXRL,174,T.NIL+T.SPOP
 5432 004106                                LIST
 5433
 5434
 5435 004106                        DOPINS  BXRL,174,T.NIL+T.SPOP
 5447 004106                                LIST
 5448
 5449
 5450
 5451
 5452
 5453 004106                        MOPINS  BXTS,200,T.NIL
 5562 004106                                LIST
 5563
 5564
 5565
 5566
 5567
 5568
 5569 004106                        MOPINS  BXWA,176,T.NIL
 5592 004106                                LIST
```

```
 5594 004106                            MOPINS  BXCR,206
 5716 004106                                    LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 78
MIXED FUNCTION          DYADIC FORMAT

   5718 004106                          MOPINS  BXEX,202
   5775 004106                                  LIST
```

```
   5777 004106                          MOPINS  BXFX,210
   5879 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 80
MIXED FUNCTION          DYADIC FORMAT

   5881 004106                              MOPINS  BXNC,204
   5950 004106                                      LIST
```

```
 5952 004106                            MOPINS  BXNL,214
 5963 004106                                    LIST
 5964
 5965
 5966 004106                            DOPINS  BXNL,214
 6104 004106                                    LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 82
MIXED FUNCTION          DYADIC FORMAT

   6106 004106                          DOPINS  FGET,150,T.DIM
   6330 004106                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 004106                          DOPINS  FPUT,152,T.DIM
   6485 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 83
MIXED FUNCTION          DYADIC FORMAT

   6487 004106                          DOPINS  FPNT,154,T.DIM
   6714 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 84
MIXED FUNCTION          DYADIC FORMAT

   6716 004106                          MOPINS  1SCN,130
   6723 004106                                  LIST
   6724
   6725
   6726 004106                          MOPINS  SCAN,62,T.DIM
   6732 004106                                  LIST
   6733
   6734
   6735 004106                          MOPINS  1RDT,124
   6741 004106                                  LIST
   6742
   6743
   6744 004106                          MOPINS  RDCT,60,T.DIM
   7044 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 85
MIXED FUNCTION          DYADIC FORMAT

   7046 004106                          MOPINS  GET!DN,0
   7193 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 86
MIXED FUNCTION          DYADIC FORMAT

   7195 004106                          MOPINS  SATAN,0
   7404 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 87
MIXED FUNCTION          DYADIC FORMAT

   7406 004106                          MOPINS  OUTR,156
   7476 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 88
MIXED FUNCTION          DYADIC FORMAT

   7478 004106                          MOPINS  DOUTER,0
   7504 004106                                  LIST
   7505
   7506
   7507
   7508
   7509
   7510 004106                          MOPINS  PRDSET,0
   7564 004106                                  LIST
   7565
   7566
   7567
   7568
   7569
   7570 004106                          MOPINS  INSPAC,0
   7612 004106                                  LIST
```

K 6

```
APLO04  MACRO M1113  08-FEB-80 09:26  PAGE 89
MIXED FUNCTION          DYADIC FORMAT

   7614 004106                          DOPINS  INNR,156
```

L 6

APL004 MACRO M1113 08-FEB-80 09:26 PAGE 90
MIXED FUNCTION   DYADIC FORMAT

```
   7960 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 91
MIXED FUNCTION          DYADIC FORMAT

   7962 004106                          MOPINS  SUBR,0
```

8725 004106 LIST

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 93
MIXED FUNCTION          DYADIC FORMAT

   8727 004106                          MOPINS  RAND,0
   8771 004106                                  LIST
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 94
MIXED FUNCTION          DYADIC FORMAT

      1         000001                          .END
```

APL004 MACRO M1113 08-FEB-80 09:26 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHRFIL= 000177          CTMAX   000212R  025 D.BXRL= 000007          D.1COM= 000001
AC1    =%000001          CHRFRO= 000135          CTRMAX  000172R  025 D.CIRC= 000001          D.1EXP= 000002
AC2    =%000002          CHRLBR= 000073              C.ALOC= 000162      D.COMB= 000001          D.1ROT= 000003
AC3    =%000003          CHRLPR= 000072              C.BLOK= 000064      D.COMP= 000001          EDICHR  005063R  025
AC4    =%000004          CHRMNS= 000137              C.BLSZ= 000140      D.CONT= 000006          EDICNT  005064R  025
AC5    =%000005          CHRMUL= 000075              C.BUFF= 177760      D.CTLM= 000000          EDIFLG  005062R  025
ALLINT  003274R     025  CHRNEG= 000100              C.BUFL= 177762      D.DEAL= 000003          EDIT   = 000003
APRNT1= ****** GX        CHRPDL= 000271              C.CCNT= 177766      D.DECD= 000002          EIS    = 000000
ARYRET  003276R     025  CHRPLU= 000055              C.CPTR= 177764      D.DIV = 000000          ELESIZ  005132R  025
ASCQUO= 000042           CHRPWR= 000120              C.CZ   = 001000     D.DMD = 000010          ENDCOR  000004R  025
A.ALEN= 000004           CHRQST= 000121              C.DIG = 000040      D.DQAD= 177777          EOF    = 104475
A.ARG = 000040           CHRQUO= 000113              C.FASW= 020000      D.DROP= 000002          ERRMAX= 000166
A.ASB = 000200           CHRRBR= 000047              C.FLEN= 000010      D.ENCD= 000002          ERRNUM  004772R  025
A.BITS= 000003           CHRRPR= 000042              C.FLSH= 004000      D.EQ   = 000000         ERRPTR  004770R  025
A.CON = 000020           CHRSEM= 000074              C.FRC = 000001      D.EXPD= 000002          ERRO    004774R  025
A.DIMS= 000002           CHRSMO= 000112              C.IDEN= 000010      D.FGET= 000005          EXEBUF  005054R  025
A.DIM1= 000006           CHRSPA= 000040              C.NAME= 000102      D.FMT = 000004          E$CT    005056R  025
A.NUM = 000006           CHRTAB= ****** GX           C.NOBF= 000400      D.FPNT= 000005          E$PT    005060R  025
A.PTR = 000000           CHRUPA= 000131              C.NUM = 000020      D.FPUT= 000005          FA.APD= 000100
A.TMP = 000010           CHR.BS= 000010              C.OP   = 000004     D.GE   = 000000         FA.CRE= 000010
BADDIR= 104463           CHR.CC= 000003              C.OPN = 010000      D.GT   = 000000         FA.DLK= 001000
BADERR= 104437           CHR.CR= 000015              C.RDLN= 177776      D.IBM = 000002          FA.ENB= 100000
BLZBUB  005116R     025  CHR.CZ= 000032              C.REWT= 002000      D.IND = 000002          FA.EXC= 002000
BRSELF  005046R     025  CHR.GT= 000046              C.RND = 000010      D.INNR= 000013          FA.EXT= 000004
CARVE = ****** GX        CHR.LF= 000012              C.SBIT= 177756      D.INTR= 000006          FA.NSP= 000100
CCNT    005072R     025  CHR.LT= 000043              C.SPFX= 000100      D.LE   = 000000         FA.NSR= 000200
CHANEX  003060R     025  CHR.SI= 000017              C.SPGL= 000200      D.LOG = 000001          FA.POS= 010000
CHAN00  000256R     025  CHR.SL= 000057              C.SPLX= 000001      D.LT   = 000000         FA.RD  = 000001
CHAN01  000330R     025  CHR.SO= 000016              C.SPRS= 000002      D.MAX = 000000          FA.RWD= 004000
CHAN02  000512R     025  CHR.TB= 000011              C.SRC  = 177774     D.MEM = 000002          FA.SEQ= 040000
CHAN03  000674R     025  CHR.UL= 000106              C.SRCP= 000024      D.MIN = 000000          FA.SHR= 000040
CHAN04  001056R     025  CHR..A= 000141              C.TT   = 000004     D.MUL = 000000          FA.TMP= 000020
CHAN05  001240R     025  CHR..E= 000145              C.WIDC= 177770      D.NE   = 000000         FA.WCK= 020000
CHAN06  001422R     025  CHR..S= 000163              C.WIDR= 177772      D.NN   = 000000         FA.WRT= 000002
CHAN07  001604R     025  CHR..T= 000164              DEBUG  = 000000     D.NR   = 000000         FD.BLK= 000010
CHAN08  001766R     025  CHR..U= 000165              DEFERR= 104405      D.OR   = 000000         FD.CCL= 000002
CHAN09  002150R     025  CHR..Z= 000172              DEVIL   005114R  025 D.PLUS= 000000          FD.COM= 020000
CHAN10  002332R     025  CHR..0= 000060              DEVNTF= 104470      D.PSUB= 000006          FD.CR  = 000002
CHAN11  002514R     025  CHR..1= 000061              DIGITS  000006R  025 D.PWR  = 000001          FD.DIR= 000010
CHAN12  002676R     025  CH.AND= 000001              DIMAX  = 000020     D.QQAD= 000000          FD.FTN= 000001
CHAN13  000022R     023  CMDBUF  003250R     025     DIMBLK  003112R  025 D.QUAD= 000000          FD.F11= 040000
CHAN14  000204R     023  CMDMAX= 000010              DIMPTR  003114R  025 D.RES  = 000000          FD.INS= 000010
CHNDPT= 104471           CMPRSS= ****** GX           DIMTYP  003116R  025 D.RHO  = 000000          FD.ISP= 002000
CHRAND= 000051           CNTSIZ  005130R     025     DIRFUL= 104466      D.ROT  = 000003         FD.MNT= 100000
CHRBEL= 000007           COMBIN  005076R     025     DOMERR= 104417      D.SCON= 000006          FD.OSP= 004000
CHRBKR= 000133           COMCNT  005126R     025     DOTRAN  000620R  020 D.SDIF= 000006          FD.PLC= 000004
CHRBOL= 000200           COMERR= 104425              DPTERR= 104416      D.SUB  = 000000         FD.PRN= 000004
CHRBOX= 000114           COMPTR  005122R     025     DTRAN1  000302R  020 D.SUBS= 000006          FD.PSE= 010000
CHRCMA= 000054           COMRTN  003156R     025     DYDCNT  003224R  025 D.TAKE= 000002          FD.RAH= 000001
CHRCOL= 000076           COMSKP  005124R     025     D$FMT   001362RG 020 D.TRAN= 000004          FD.RAN= 000002
CHRDEL= 000107           CORPTR  005120R     025     D$TRAN  000000RG 020 D.UNON= 000006          FD.REC= 000001
CHRDIV= 000053           CS.ASS= 000200              D.AND  = 000000     D.$MAX= 000000          FD.RWM= 000001
CHRDLR= 000176           CS.BOO= 000010              D.BXCT= 000007      D.$MIN= 000000          FD.SDI= 000020
CHRDLT= 000110           CS.FLO= 000020              D.BXIO= 000007      D.$MUL= 000000          FD.SQD= 000040
CHRDOT= 000056           CS.IDN= 000040              D.BXNL= 000007      D.$PLU= 000000          FD.TTY= 000004
CHREP  = 000105          CS.ONE= 000100              D.BXPP= 000007      D.$RES= 000000          FD.WBH= 000002
CHRERR= 104431           CTEPS   000152R     025     D.BXPW= 000007      D.$SUB= 000000          FF.CHR= 000005
```

APLOO4 MACRO M1113 08-FEB-80 09:26 PAGE 94-2
SYMBOL TABLE

```
FF.NV = 000003          F.DSPT= 000044          F.TRC = 000002          LININC  005026R 025 M.IBM = 000003
FF.POE= 000002          F.DVNM= 000134          F.UNIT= 000136          LISIZE= 000020      M.IDN = 000000
FF.RWD= 000001          F.EFBK= 000010          F.URBD= 000020          LITOP   005072R 025 M.INSP= 000006
FF.RWF= 000006          F.EFN = 000050          F.VAL = 177776          LNLNG = 104541      M.IOTA= 000000
FF.SPC= 000004          F.END = 002000          F.VBN = 000064          LOCLCT  005036R 025 M.LIT = 000205
FHEADR  005012R 025     F.EOBB= 000032          F.VBSZ= 000060          LOG     003562R 020 M.LN  = 000001
FHRET   005040R 025     F.ERR = 000052          GETCOR= ****** GX       LOWCOR  000002R 025 M.MMD = 000010
FILL    003372R 020     F.FACC= 000043          GETFRA= ****** GX       LSTERP  004766R 025 M.NEG = 000000
FILNTF= 104467          F.FFBY= 000014          GETREG= ****** GX       LWIDTH  000070R 025 M.NOT = 000000
FIS   = 000000          F.FLAG= 000002          GNOME   003102R 025 L.BITS= 000003      M.NUB = 000006
FISCON= 046000          F.FLB = 000100          HASHC   005136R 025 L.CNT = 000002      M.NUM = 000204
FIXPTR  003074R 025     F.FNAM= 000110          HASHTB  005140R 025 L.PRNT= 000001      M.OUTR= 000012
FLAGS   004764R 025     F.FNB = 000102          HSHMSK= 177601      L.PTR = 000000      M.PITM= 000000
FLCBUF  003250R 025     F.FNDF= 000010          HSHSIZ= 000200      L.PTR1= 000004      M.PRDS= 000006
FLINE   005014R 025     F.FTYP= 000116          IASH  = ****** GX   L.SUBS= 000002      M.QQAD= 000000
FLOEXP  003270R 025     F.FUN = 000010          ICHAN   005050R 025 MACHIN= 000106      M.QUAD= 000000
FLONEG  003272R 025     F.FVER= 000120          IDIV  = ****** GX   MAXCHN= 000014      M.RAND= 000012
FLSWTB  000222R 025     F.HEAD= 000006          ILARGC= ****** GX   MONBLK  003120R 025 M.RAVL= 000000
FLTLEN= 000010          F.HIBK= 000004          ILFNAM= 104464      MONCVT  003132R 025 M.RCIP= 000000
FMTDOM  002704R 020     F.LBL = 000020          ILLCHN= 104540      MONLEN  003130R 025 M.RDCT= 000011
FMTRND  003454R 020     F.LINE= 000012          ILLSW = 104565      MONPTR  003122R 025 M.REVS= 000003
FNINSI  005044R 025     F.LOC = 000004          IMUL  = ****** GX   MONSFT  003126R 025 M.ROLL= 000000
FNLIST= ****** GX       F.LSTC= 000004          INDERR= 104410      MONTYP  003124R 025 M.SATA= 000001
FORIGN  000032R 025     F.LUN = 000042          INNTMP  003250R 025 MONVAL  003134R 025 M.SCAN= 000011
FO.APD= 000106          F.MBCT= 000054          INNTM2  005016R 025 M$ENCD  000766RG 020 M.SHAP= 000000
FO.MFY= 000002          F.MBC1= 000055          INPTR   000064R 025 M$FMT   001332RG 020 M.SIGN= 000000
FO.RD = 000001          F.MBFG= 000056          INSNAM= 104462      M.ABS = 000000      M.SUBR= 000001
FO.UPD= 000006          F.MON = 000002          IOERR = 104477      M.BOL = 000200      M.SYM = 000203
FO.WRT= 000016          F.NIL = 000001          I.CNT = 177776      M.BXAV= 000007      M.SYN = 000201
FPMODE  000042R 025     F.NRBD= 000024          I.DIM = 177772      M.BXCR= 000007      M.TRAN= 000012
FPP   = 000000          F.NREC= 000030          I.LEN = 000012      M.BXCT= 000007      M.$ABS= 000000
FPPLEN= 000004          F.OPN = 000010          I.OERR= 000062      M.BXEX= 000007      M.$CEI= 000000
FRARGC= ****** GX       F.OVBS= 000030          I.OLD = 177770      M.BXFX= 000007      M.$FLR= 000000
FREBLK  007400R 025     F.PREV= 000000          I.PTR = 177774      M.BXIO= 000007      M.$IDN= 000000
FUZZ    000142R 025     F.PRNT= 100000          I.STR = 177766      M.BXLC= 000003      M.$NEG= 000000
FXPTR1  003076R 025     F.PROT= 010000          KILL  = ****** GX   M.BXNC= 000007      M.$SIG= 000000
F$EVAL= 000100          F.RACC= 000016          LABERR= 104406      M.BXNL= 000007      M.1RDT= 000011
F$EVI = 000200          F.RATT= 000001          LABLPT  005042R 025 M.BXPP= 000007      M.1REV= 000003
F$EXEC= 000400          F.RCNM= 000034          LARGBL  003162R 025 M.BXPW= 000007      M.1SCN= 000011
F$FUNC= 001000          F.RCTL= 000017          LARGCV  003176R 025 M.BXRL= 000007      NAMBUF  003240R 025
F.ACTL= 000076          F.READ= 004000          LARGFL  003160R 025 M.BXTS= 000007      NB.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LARGLE  003174R 025 M.BXWA= 000007      NB.DIR= 000100
F.BBFS= 000062          F.RES = 000040          LARGLN  003166R 025 M.CEI = 177777      NB.NAM= 000004
F.BDB = 000070          F.RET = 177772          LARGPT  003164R 025 M.CHR = 000202      NB.SD1= 000400
F.BGBC= 000057          F.RSIZ= 000002          LARGSF  003172R 025 M.CMT = 000206      NB.SD2= 001000
F.BITS= 177774          F.RTYP= 000000          LARGTY  003170R 025 M.DOUT= 000006      NB.SNM= 000040
F.BKDN= 000026          F.R2  = 000014          LARGVA  003200R 025 M.DQAD= 000006      NB.STP= 000020
F.BKDS= 000020          F.R3  = 000016          LBINC   005100R 025 M.ENCD= 000004      NB.SVR= 000010
F.BKEF= 000050          F.R4  = 000020          LBUFF   003316R 025 M.EXE = 000000      NB.TYP= 000002
F.BKP1= 000051          F.SEQN= 000100          LBUFLN= 000621      M.EXP = 000001      NB.VER= 000001
F.BKST= 000024          F.SPDV= 000072          LCOUNT  005106R 025 M.FACT= 000001      ND.DEV= 000200
F.BKVB= 000064          F.SPUN= 000074          LENERR= 104412      M.FIL = 000177      ND.EXT= 000002
F.BR  = 000020          F.STBK= 000036          LENTAB  000054R 025 M.FLR = 177777      ND.NAM= 000004
F.CHR = 000075          F.STK = 000022          LEXPTR  003100R 025 M.FMT = 000004      ND.PPN= 000100
F.CNTG= 000034          F.STOP= 000004          LFTOP   005066R 025 M.GDUP= 000003      ND.SEX= 000000
F.CODE= 000010          F.STP = 000001          LINC    005102R 025 M.GDWN= 000003      ND.SES= 000020
F.DFNB= 000046          F.SUSP= 000002          LINE    005016R 025 M.GETI= 000006      ND.SNM= 000000
```

APL004 MACRO M1113 08-FEB-80 09:26 PAGE 94-3
SYMBOL TABLE

```
ND.$NS= 000040          OPTMP5  005100R     025 R.BITS= 000003          SY.BAD= 000001          T.SOP = 000002
ND.$PN= 001400          OPTMP6  005102R     025 R.FIX = 000001          SY.BRK= 100000          T.SOPP= 000040
NEGSGN  005063R     025 OPTMP7  005104R     025 R.LCT = 000002          SY.BS = 000010          T.TMP = 000200
NINPT1  003302R     025 OPTMP8  005106R     025 R.LN  = 000002          SY.CLR= 000040          T.VAL = 100000
NINPT2  003300R     025 OPTMP9  005110R     025 R.LNUM= 000010          SY.ERP= 000020          T.VAR = 001000
NINSTK  003304R     025 OPTM10  005112R     025 R.NEXT= 000006          SY.NOU= 000100          UPPER = 177400
NINTMP  003310R     025 OPTM14  005026R     025 R.OFF = 000000          SY.OPN= 000002          UTNAME  000344R     023
NINTYP  003306R     025 ORIGIN  000030R     025 R.PREV= 000004          SY.PMT= 001000          VALERR= 104413
NORM    002706R     020 OUTPTR  000066R     025 R.PTR = 000000          SY.SRT= 000400          VARBLE  007400R     025
NOTAVL= 104472          OVROPR= 000004          R.SEQ = 000003          SY.SVC= 000004          VARBLS  000000R     025
NOTOPN= 104473          PATCH   007344R     025 R.VAR = 000002          SY.SVL= 000200          VERNUM  000000R     025
NPRT1   004762R     025 PRECSN  000052R     025 R6    =%000006          S.AB  = 177776          VRSION= 000002
NULL    007364R     025 RANDAC  003064R     025 R7    =%000007          S.ASB = 100000          WIDTH   000072R     025
NUMCHN= 000020          RANDY   003062R     025 SAME  = ****** GX       S.BFHD= 000020          WROUND  000020R     025
NUMDFL  003230R     025 RANERR= 104411          SCANCL  002512R     020 S.BITS= 000002          WSDATE  000136R     025
NUMEFL  003233R     025 RARGBL  003204R     025 SCI     003746R     020 S.CHR1= 000003          WSFULL= 104403
NUMESN  003236R     025 RARGCV  003220R     025 SETBIG= ****** GX       S.FATT= 000016          WSNAME  000074R     025
NUMFLG  003230R     025 RARGFL  003202R     025 SI    =%000005          S.FDB = 000140          WSTIME  000132R     025
NUMLEN  000062R     025 RARGLE  003216R     025 SIDAMN= 104433          S.FLB = 040000          $ADDF = ****** GX
NUMNEG  003232R     025 RARGLN  003210R     025 SIFRAM  003314R     025 S.FNAM= 000006          $CFCC = ****** GX
NUMQFL  003234R     025 RARGPT  003206R     025 SISTAK  006350R     025 S.FNB = 000036          $CMPF = ****** GX
NUMRIG  003231R     025 RARGSF  003214R     025 SITOP   006346R     025 S.FNBW= 000017          $DIVF = ****** GX
NUMWID  003226R     025 RARGTY  003212R     025 SITP    007144R     025 S.FNTY= 000004          $FIS  = ****** G
NUMZFL  003235R     025 RARGVA  003222R     025 SIWARN= 000200          S.FTYP= 000002          $FPP  = ****** G
NXTIND= ****** GX       RBASE   005074R     025 SIZTAB  000044R     025 S.LEN = 000002          $FPPLN= ****** G
NYIERR= 104432          RCOUNR  005112R     025 SI.BIT= 177774          S.LINK= 177774          $LDCDF= ****** GX
N.BLSZ= 000036          RCOUNT  005110R     025 SI.LEN= 177772          S.NFEN= 000020          $LDCFD= ****** GX
N.DESC= 000020          REDCNT  005070R     025 SI.RET= 177772          S.TYPE= 140000          $LDF  = ****** GX
N.DEV = 000032          REDIGN  005066R     025 SI.VAL= 177776          S.VAL = 000000          $MACHN= ****** G
N.DEVN= 000034          REDOFF  005072R     025 SKPSIZ  005134R     025 TKDIMS  005072R     025 $MODF = ****** GX
N.DID = 000024          REENTR  003060R     025 SPHOLD  000004R     024 TKFRAM  005066R     025 $MULF = ****** GX
N.DVNM= 000032          REGTYP= 000002          SPINIT  000002R     024 TKLEFT  005070R     025 $NEGF = ****** GX
N.EXT = 000014          RESBIT  003142R     025 STKINC= 000774          TMODE   000403R     023 $RSTS = ****** G
N.FID = 000000          RESCVT  003152R     025 STOP  = 104420          TMPTAG= 000166          $RSX  = ****** G
N.FNAM= 000006          RESERR= 104440          SUBCNT  003104R     025 TTYBUF  004140R     025 $RT11 = ****** G
N.FTYP= 000014          RESLEN  003150R     025 SUBDCT  003106R     025 TTYLEN= 000622          $SETD = ****** GX
N.FVER= 000016          RESPTR  003140R     025 SUBPTR  003110R     025 TURKEY= 104436          $SETF = ****** GX
N.NAME= 000006          RESSFT  003146R     025 SYMBOL= ****** GX       TYPE  = 000166          $STCFI= ****** GX
N.NEXT= 000022          RESTYP  003144R     025 SYMB1   005344R     025 T.BOL = 020000          $STEXP= ****** GX
N.PPN = 000024          RESULT  003136R     025 SYMF    005340R     025 T.CON = 004000          $TSTF = ****** GX
N.STAT= 000020          RESVAL  003154R     025 SYMPT1  005346R     025 T.DIM = 000100          .BOOLE= 000002
N.UNIT= 000034          RINC    005104R     025 SYMTAB  005352R     025 T.FUN = 010000          .CHARA= 000000
OBUFLN= 001000          RLTBAS  005066R     025 SYMTBF= 104404          T.LIST= 000400          .DEF. = 000002
OCHAN   005052R     025 ROTINC  005070R     025 SYMTOP  005350R     025 T.LIT = 000020          .FLOAT= 000006
OMODE   000402R     023 ROUND   000010R     025 SYMTP   006272R     025 T.MON = 000010          .FSRCB= ****** G
OPTMP0  005066R     025 ROVER   003312R     025 SYMWRN= 000054          T.NIL = 000004          .INTEG= 000004
OPTMP1  005070R     025 RSTS  = 000000          SYNERR= 104407          T.NUM = 002000          .MOLUN  000404R     023
OPTMP2  005072R     025 RSX   = 000115          SYSER = ****** GX       T.OPFN= 000001          ...GBL= 000000
OPTMP3  005074R     025 RTOP    005070R     025 SYSERR= 104402          T.OTHR= 040000          ...TPC= 000020
OPTMP4  005076R     025 RT11  = 000000          SYSTAT  000000R     024

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
```

```
APL004  MACRO M1113  08-FEB-80 09:26  PAGE 94-4
SYMBOL TABLE

MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    004106      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13824 WORDS  ( 54 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:52
APL004,DM1:[305,301]APL004/-SP=APLRX1,APLDEF,APL004,APLOP,END
```

FILEID**APL005

```
  AAAAAA      PPPPPPPP      LL              000000        000000     5555555555
  AAAAAA      PPPPPPPP      LL              000000        000000     5555555555
AA      AA    PP      PP    LL            00    00    00    00     55
AA      AA    PP      PP    LL            00    00    00    00     55
AA      AA    PP      PP    LL            00    00    00  0000     555555
AA      AA    PP      PP    LL            00    00    00  0000     555555
AA      AA    PPPPPPPP      LL            00    00    00  00  00         55
AA      AA    PPPPPPPP      LL            00    00    00  00  00         55
AAAAAAAAAA    PP            LL            00    00    0000    00           55
AAAAAAAAAA    PP            LL            00    00    0000    00           55
AA      AA    PP            LL            00    00    00      00     55    55      ....
AA      AA    PP            LL            00    00    00      00     55    55      ....
AA      AA    PP            LLLLLLLLLL      000000        000000       555555      ....
AA      AA    PP            LLLLLLLLLL      000000        000000       555555      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 09:29
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS                RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    82-  6107    MIXED FUNCTION             FILE INPUT
```

APL005 MACRO M1113 08-FEB-80 09:29
TABLE OF CONTENTS

82- 6337   MIXED FUNCTION   FILE OUTPUT
83- 6488   MIXED FUNCTION   SET FILE POINTER

```
      1        000000                  RT11    =  0          ;NOT RT-11
      2        000000                  RSTS    =  0          ;NOT RSTS/E
      3        000115                  RSX     = 'M          ;RSX-11/M
      4
      5        000000                  EIS     =  0          ;NO EIS
      6        000000                  FIS     =  0          ;NO FIS
      7        000000                  FPP     =  0          ;NO FPP
      8
      9        000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\

```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  APLOO5
                                .IDENT  \V02.03\
                                
                                ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
                                ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
                                ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
                                ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
                                ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO5 IS:
                                ;
                                ;       APLOO5 < ...,APLOO5,APLOP
                                ;

000005                          OVROPR= 5                       ;OVERLAY SPECIFIED OPERATORS
```

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                                ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                                ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                                ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                                ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                                ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                                ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                                ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                                ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                                ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                                ;       INTACT!
                                ;--



                                ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                                ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                                ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                                ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                                ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                                ;MACROS.

                                .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-D.'NAME
                                        .PSECT  CODE
                                D$'NAME::
                                .ENDM

                                .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                                .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                                .IF EQ OVROPR-M.'NAME
                                        .PSECT  CODE
                                        .IF NE NUM
                                M$'NAME::
                                    .ENDC
                                .ENDM

                                .MACRO  LIST
                                        .ENDC
                                .ENDM

                                .MACRO  PRMRES
                                        .IF EQ 1
                                .ENDM

                                                ;*********** ASSEMBLING THE FILE!!!!!
160 007400                      PRMRES
199 007400                              LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,,FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

DEFINITIONS   ERRORS

```
    641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
    652 007400                                  LIST
    653
    654
    655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
    662 007400                                  LIST
    663
    664
    665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
    676 007400                                  LIST
    677
    678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
    687 007400                                  LIST
    688
    689
    690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
    700 007400                                  LIST
    701
    702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    712 007400                                  LIST
    713
    714
    715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    733 007400                                  LIST
    734
    735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    745 007400                                  LIST
    746
    747
    748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
    758 007400                                  LIST
    759
    760
    761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
    780 007400                                  LIST
    781
```

```
    783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    786 007400                                  LIST
    787
    788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    791 007400                                  LIST
    792
    793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
    799 007400                                  LIST
    800
    801
    802 007400                          MOPINS  $IDN,-2
    808 007400                                  LIST
    809
    810
    811 007400                          MOPINS  $FLR,-14
    855 007400                                  LIST
    856
    857
    858 007400                          MOPINS  $CEI,-16
    868 007400                                  LIST
    869
    870
    871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
    878 007400                                  LIST
    879
    880 007400                          MOPINS  $NEG,-4
    887 007400                                  LIST
    888
    889
    890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
    901 007400                                  LIST
    902
    903 007400                          MOPINS  $SIG,-6
    914 007400                                  LIST
    915
    916
    917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
    928 007400                                  LIST
    929
    930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
    939 007400                                  LIST
    940
    941
    942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
    949 007400                                  LIST
    950
    951
    952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
    959 007400                                  LIST
    960
    961 007400                          PRMRES
    966 007400                                  LIST
    967
    968
    969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
    977 007400                                  LIST
    978
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,.FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 51
DEFINITIONS     ERRORS

   1189 007400                          MOPINS  SHAP,102
   1211 00740C                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

DEFINITIONS ERRORS

```
1315 007400                          MOPINS  RAVL,56
1325 007400                                  LIST
1326
1327
1328
1329
1330
1331 007400                          DOPINS  CTLM,56,T.DIM
1596 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

DEFINITIONS ERRORS

```
1701 007400                          DOPINS  1COM,124
1709 007400                                  LIST
1710
1711
1712
1713
1714
1715 007400                          DOPINS  COMP,60,T.DIM
1789 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

DEFINITIONS ERRORS

```
1866 007400                          DOPINS  DROP,104
1907 007400                                  LIST
1908
1909
1910
1911 007400                          DOPINS  TAKE,114
2149 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 57
DEFINITIONS     ERRORS

   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```

```
 2206 007400                            DOPINS  1ROT,146
 2214 007400                                    LIST
 2215
 2216
 2217
 2218
 2219
 2220 007400                            DOPINS  ROT,144,T.DIM
 2328 007400                                    LIST
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 60
DEFINITIONS     ERRORS

   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 61
DEFINITIONS     ERRORS

   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 62
DEFINITIONS     ERRORS

   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 64
DEFINITIONS     ERRORS

   3221 007400                          MOPINS  MMD,122
   3316 007400                                  LIST
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 65
DEFINITIONS     ERRORS

   3318 007400                          DOPINS  DMD,122
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 66
DEFINITIONS     ERRORS

   3864 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 67
DEFINITIONS     ERRORS

   3866 007400                          DOPINS  MEM,70
   3921 007400                                  LIST
```

```
   3923 007400                          MOPINS  NUB,106
   3986 007400                                  LIST
   3987
   3988
   3989
   3990
   3991
   3992 007400                          DOPINS  UNON,106
   4022 007400                                  LIST
   4023
   4024
   4025
   4026
   4027
   4028 007400                          DOPINS  INTR,66
   4042 007400                                  LIST
   4043
   4044
   4045
   4046
   4047
   4048 007400                          DOPINS  SDIF,50
   4161 007400                                  LIST
```

```
   4163 007400                          DOPINS  SCON,234
   4179 007400                                  LIST
   4180
   4181
   4182 007400                          DOPINS  CONT,112
   4200 007400                                  LIST
   4201
   4202
   4203 007400                          DOPINS  PSUB,236
   4209 007400                                  LIST
   4210
   4211
   4212 007400                          DOPINS  SUBS,116
   4288 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 70
DEFINITIONS     ERRORS

   4290 007400                          MOPINS  EXE,70
   4336 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 71
DEFINITIONS     ERRORS

   4338 007400                          MOPINS  ENCD,76
   4411 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 72
DEFINITIONS     ERRORS

   4413 007400                          MOPINS  FMT,160
   4423 007400                                  LIST
   4424
   4425
   4426
   4427
   4428
   4429 007400                          DOPINS  FMT,160
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 73
DEFINITIONS     ERRORS

   4953 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 74
DEFINITIONS     ERRORS

   4955 007400                          MOPINS  IBM,132,,,MONIBM
   4958 007400                                  LIST
   4959
   4960
   4961 007400                          DOPINS  IBM,132
   5046 007400                                  LIST
```

```
 5048 007400                            MOPINS  QUAD,74,T.NIL+T.SPOP
 5075 007400                                    LIST
 5076
 5077
 5078
 5079
 5080
 5081 00740C                            DOPINS  QUAD,74,T.NIL+T.SPOP
 5097 007400                                    LIST
 5098
 5099
 5100
 5101
 5102
 5103 007400                            MOPINS  QQAD,142,T.NIL+T.SPOP
 5135 007400                                    LIST
 5136
 5137
 5138
 5139
 5140
 5141 007400                            DOPINS  QQAD,142,T.NIL+T.SPOP
 5152 007400                                    LIST
 5153
 5154
 5155
 5156
 5157
 5158 007400                            MOPINS  DQAD,134,T.NIL+T.SPOP
 5192 007400                                    LIST
 5193
 5194
 5195
 5196 007400                            DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
 5199 007400                                    LIST
```

```
5201 007400                         MOPINS  BXAV,162,T.NIL
5219 007400                                 LIST
5220
5221
5222
5223
5224
5225 007400                         MOPINS  BXCT,164,T.NIL+T.SPOP
5234 007400                                 LIST
5235
5236
5237 007400                         DOPINS  BXCT,164,T.NIL+T.SPOP
5272 007400                                 LIST
5273
5274
5275
5276
5277
5278 007400                         MOPINS  BXIO,166,T.NIL+T.SPOP
5287 007400                                 LIST
5288
5289
5290 007400                         DOPINS  BXIO,166,T.NIL+T.SPOP
5302 007400                                 LIST
5303
5304
5305
5306
5307 007400                         MOPINS  BXLC,212,T.NIL
5340 007400                                 LIST
5341
5342
5343
5344
5345
5346
5347 007400                         MOPINS  BXPP,170,T.NIL+T.SPOP
5356 007400                                 LIST
5357
5358
5359 007400                         DOPINS  BXPP,170,T.NIL+T.SPOP
5381 007400                                 LIST
5382
5383
5384
5385
5386
5387
5388 007400                         MOPINS  BXPW,172,T.NIL+T.SPOP
5397 007400                                 LIST
5398
5399
5400 007400                         DOPINS  BXPW,172,T.NIL+T.SPOP
5417 007400                                 LIST
5418
5419
5420
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 76-1
DEFINITIONS     ERRORS

   5421
   5422
   5423 007400                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 007400                                  LIST
   5433
   5434
   5435 007400                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 007400                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 007400                          MOPINS  BXTS,200,T.NIL
   5562 007400                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 007400                          MOPINS  BXWA,176,T.NIL
   5592 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 77
DEFINITIONS     ERRORS

   5594 007400                          MOPINS  BXCR,206
   5716 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 78
DEFINITIONS     ERRORS

   5718 007400                          MOPINS  BXEX,202
   5775 007400                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 79
DEFINITIONS     ERRORS

   5777 007400                          MOPINS  BXFX,210
   5879 007400                                  LIST
```

M 10

APLOO5 MACRO M1113 08-FEB-80 09:29 PAGE 80
DEFINITIONS   ERRORS

```
   5881 007400                          MOPINS  BXNC,204
   5950 007400                                  LIST
```

N 10

```
   5952 007400                          MOPINS  BXNL,214
   5963 007400                                  LIST
   5964
   5965
   5966 007400                          DOPINS  BXNL,214
   6104 007400                                  LIST
```

```
                                        DOPINS  FGET,150,T.DIM
                                        .SBTTL  MIXED FUNCTION          FILE INPUT

                                        ;++
                                        ; READS INFORMATION FROM A FILE OPEN ON THE SPECIFIED CHANNEL.
                                        ;THE FORMAT IS: CHAN .IQ [TYPE] N
                                        ;WHERE CHAN IS THE CHANNEL NUMBER, .IQ IS THE FILE INPUT OPERATOR,
                                        ;[TYPE] IS THE FILE TYPE (SEE BRANCH TABLE BELOW), AND N IS THE NUMBER
                                        ;OF COMPONENTS OF THE INDICATED TYPE TO READ.
                                        ;--

                                                .DSABL LSB
000000                                          CALL    FSETUP          ;SET UP FILE POINTERS
000004                                          PUSH    ICHAN           ;SAVE CURRENT INPUT CHANNEL
000010  010237  005050'                        MOV     R2,ICHAN        ;AND MAKE FILE THE CURRENT INPUT CHANNEL
000014  013701  003212'                        MOV     RARGTY,R1       ;RIGHT ARG NUMERIC?
000020  001003                                BNE     2$              ;YES
000022  104417                        1$:     DOMERR                  ;BAD COMPONENT COUNT
000024  104412                        28$:    LENERR                  ;COMPONENT COUNT NOT A SINGLETON ARRAY
000026  104565                        29$:    ILLSW                   ;Wrong file format
000030  022764  000001  000004      2$:     CMP     #1,A.ALEN(R4)   ;SCALAR?
000036  001372                                BNE     28$             ;NO
000040  016137  000000G 003220'              MOV     IRARGC(R1),RARGCV ;FETCH ROUTINE FOR LENGTH (INTEGER)
000046                                          PUSH    R3              ;SAVE FILE TYPE
000050  011403                                MOV     (R4),R3         ;POINT TO DATA
000052                                          CALL    @RARGCV         ;FETCH ARG
000056  013704  003222'                        MOV     RARGVA,R4       ;GET # OF ELEMENTS
000062  003757                                BLE     1$              ;TOO SMALL
000064  042762  001000  177756                BIC     #C.CZ,C.SBIT(R2);DON'T WRITE CNTL-Z @ EOF
000072                                          POP     R3              ;ASCII SEQUENTIAL FILE?
000074  001033                                BNE     4$              ;NO
                                        ;
                                        ;ASCII SEQUENTIAL FILE TYPE.  READ A LINE FROM THE FILE.
                                        ;
000076  032762  020000  177756                BIT     #C.FASW,C.SBIT(R2);Was the file opened Formatted ASCII?
000104  001750                                BEQ     29$             ;No, Can't do this then
000106  052737  000100  000000'              BIS     #SY.NOU,SYSTAT  ;PREVENT OUTPUT
000114                                          PUSH    <R2,R3,R4>      ;SAVE SOME REGS
000122                                          CALL    GETFA           ;GET A LINE OF INPUT
000126                                          CALL    FIXLIN          ;FIX IT
000132  122744  000015                        CMPB    #CHR.CR,-(R4)   ;IS LAST CHAR A CR?
000136  001001                                BNE     3$              ;NO
000140  005301                                DEC     R1              ;YES, DON'T INCLUDE IT IN STRING
000142                                  3$:     POP     <R4,R3,R2>      ;RESTORE REGS
000150  042737  000100  000000'              BIC     #SY.NOU,SYSTAT  ;OUTPUT ALLOWED AGAIN
000156                                          PUSH    R0              ;SAVE POINTER TO LINE BUFFER
000160  010104                                MOV     R1,R4           ;COPY DATA LENGTH
000162  000404                                BR      5$              ;Go get block for result
                                        ;
                                        ; ALLOCATE BLOCK FOR RESULT
                                        ;
000164                                  4$:
000164  032762  020000  177756                BIT     #C.FASW,C.SBIT(R2);Was the file opened Formatted ASCII?
000172  001315                                BNE     29$             ;Yes, Wrong file format
000174  010401                          5$:     MOV     R4,R1           ;COPY DATA LENGTH TO READ
000176  016300  000000'                        MOV     DATATY(R3),R0   ;GET TYPE OF RESULT
000202  010037  003144'                        MOV     R0,RESTYP       ;SAVE RESULT TYPE
```

```
6178 000206                                     ASH     SIZTAB(R0),R1   ;COMPUTE SIZE IN BYTES
6179 000222  012700  000010                     MOV     #A.DIM1+2,R0    ;HEADER SIZE
6180 000226                                     PUSH    R0              ;SAVE AS OFFSET
6181 000230  060100                             ADD     R1,R0           ;TOTAL LENGTH OF DATA
6182 000232                                     CALL    GETFRA          ;Get result, SI frame for it
6183 000236  062620                             ADD     (SP)+,(R0)+     ;ADD OFFSET TO DATA
6184 000240  112720  000001                     MOVB    #1,(R0)+        ;1 DIMENSION
6185 000244  113710  003144'                    MOVB    RESTYP,(R0)     ;BITS
6186 000250  001003                             BNE     FGET1           ;WAS NUMERIC
6187 000252  042765  002000  177774             BIC     #T.NUM,SI.BIT(SI);BAD GUESS
6188 000260  152720  000200           FGET1:   BISB    #A.ASB,(R0)+    ;FLAG ARRAY
6189 000264  010420                             MOV     R4,(R0)+        ;ALEN
6190 000266  010420                             MOV     R4,(R0)+        ;DIM1 (VECTOR)
6191 000270  060307                             ADD     R3,PC           ;AND BRANCH TO CORRECT FILE TYPE
6192                                     ;
6193                                     ;BRANCHES TO DIFFERENT FILE TYPE ROUTINES
6194                                     ;
6195 000272  000405                             BR      6$              ; 0 -- ASCII SEQUENTIAL (LINE ORIENTED)
6196 000274  000413                             BR      9$              ; 1 -- BYTE VALUES
6197 000276  000423                             BR      11$             ; 2 -- INTEGER VALUES
6198 000300  000434                             BR      13$             ; 3 -- APL INTERNAL LITERAL FORMAT
6199 000302  000442                             BR      15$             ; 4 -- SINGLE PRECISION FLOATING
6200 000304  000466                             BR      17$             ; 5 -- DOUBLE PRECISION FLOATING
6201
6202                                     ;
6203                                     ;ASCII SEQUENTIAL
6204                                     ;
6205 000306                      6$:     POP     R3              ;GET POINTER TO LINE BUFFER
6206 000310  005704                             TST     R4              ;ANYTHING?
6207 000312  001403                             BEQ     8$              ;NULL LINE
6208 000314  112320              7$:     MOVB    (R3)+,(R0)+     ;COPY 1 BYTE
6209 000316                             SOB     R4,7$           ;COPY ENTIRE LINE
6210 000322  000504              8$:     BR      20$             ;ALL DONE
6211                                     ;
6212                                     ;BYTE VALUES
6213                                     ;
6214 000324  010003              9$:     MOV     R0,R3           ;COPY OUTPUT POINTER
6215 000326                      10$:    CALL    GETBYT          ;GET A BYTE
6216 000332  042700  177400             BIC     #UPPER,R0       ;CLEAR GARBAGE BITS IF ANY
6217 000336  010023                             MOV     R0,(R3)+        ;STORE RESULT
6218 000340                             SOB     R4,10$          ;FLOAT ALL BYTES
6219 000344  000473                             BR      20$
6220                                     ;
6221                                     ;INTEGER VALUES
6222                                     ;
6223 000346  010003              11$:    MOV     R0,R3           ;COPY OUTPUT POINTER
6224 000350                      12$:    CALL    GETBYT          ;GET A BYTE FROM THE FILE
6225 000354  110023                             MOVB    R0,(R3)+        ;SAVE IT
6226 000356                             CALL    GETBYT          ;GET ANOTHER BYTE
6227 000362  110023                             MOVB    R0,(R3)+        ;STORE IT
6228 000364                             SOB     R4,12$          ;LOOP
6229 000370  000461                             BR      20$
6230                                     ;
6231                                     ;APL INTERNAL LITERAL
6232                                     ;
6233 000372  010003              13$:    MOV     R0,R3           ;COPY OUTPUT POINTER
6234 000374                      14$:    CALL    GETBYT          ;GET A BYTE
```

```
6235 000400  110023                        MOVB    R0,(R3)+        ;STORE IT IN RESULT
6236 000402                                SOB     R4,14$          ;THAT'S EASY!
6237 000406  000452                        BR      20$
6238                                ;
6239                                ;SINGLE PRECISION FLOATING
6240                                ;
6241 000410                        15$:    PUSH    R4              ;SAVE COUNT
6242 000412  010003                        MOV     R0,R3           ;COPY RESULT POINTER
6243 000414  012704  000004        16$:    MOV     #4,R4           ;COUNT OF DATA IN FILE
6244 000420                        22$:    CALL    GETBYT          ;GET A BYTE OF DATA
6245 000424  110023                        MOVB    R0,(R3)+        ;STORE IT
6246 000426                                SOB     R4,22$          ;LOOP FOR DATA
6247 000432  013704  000062'               MOV     NUMLEN,R4       ;GET LENGTH OF DATA
6248 000436  162704  000004                SUB     #4,R4           ;  MINUS LENGTH OF INPUT
6249 000442  001403                        BEQ     24$             ;NOTHING MORE TO DO
6250 000444  105023                23$:    CLRB    (R3)+           ;CLEAR AWAY LEFT-OVERS
6251 000446                                SOB     R4,23$          ;LOOP FOR REMAINING SPACE
6252 000452  005316                24$:    DEC     (SP)            ;DECREMENT COUNT
6253 000454  003357                        BGT     16$             ;LOOP UNTIL DONE
6254 000456  005726                        TST     (SP)+           ;POP COUNT
6255 000460  000425                        BR      20$
6256                                ;
6257                                ;DOUBLE PRECISION FLOATING
6258                                ;
6259 000462                        17$:    PUSH    R4              ;SAVE COUNT
6260 000464  010003                        MOV     R0,R3           ;SHIFT POINTER TO DATA
6261 000466  013704  000062'       18$:    MOV     NUMLEN,R4       ;GET SIZE OF FP NUMBER
6262 000472                        25$:    CALL    GETBYT          ;GET A BYTE
6263 000476  110023                        MOVB    R0,(R3)+        ;STORE IT
6264 000500                                SOB     R4,25$          ;LOOP FOR ALL CHARACTERS
6265 000504  012704  000010                MOV     #8.,R4          ;LENGTH OF DATA TO FETCH
6266 000510  163704  000062'               SUB     NUMLEN,R4       ;MINUS FP NUMBER LEN
6267 000514  001404                        BEQ     27$             ;NOTHING TO SAY, NOTHING TO DO
6268 000516                        26$:    CALL    GETBYT          ;GET BYTE AND IGNORE
6269 000522                                SOB     R4,26$          ;LOOP FOR UNWANTED STUFF
6270 000526  005316                27$:    DEC     (SP)            ;DECREMENT COUNT
6271 000530  003356                        BGT     18$             ;LOOP UNTIL DONE
6272 000532  005726                        TST     (SP)+           ;POP COUNT
6273
6274 000534                        20$:    POP     ICHAN           ;RESTORE INPUT CHANNEL
6275 000540                                RETURN                  ;AND RETURN
6276
6277                                ;++
6278                                ; FUNCTIONAL DESCRIPTION:
6279                                ;       FSETUP SETS UP POINTERS FOR THE FILE OPERATORS.
6280                                ;
6281                                ; OUTPUTS:
6282                                ;       R1      - TYPE FOR RIGHT ARG
6283                                ;       R2      - CHANNEL BLOCK POINTER
6284                                ;       R3      - FILE TYPE
6285                                ;       R4      - RIGHT ARG BLOCK POINTER
6286                                ;--
6287
6288         000005                 MAXTYP= 5                       ;MAX FILE TYPE
6289
6290 000542  012701  000004        FSETUP: MOV     #.INTEGER,R1    ;INTEGER INTERMEDIATE
6291 000546                                CALL    GETSCA          ;GET SCALAR
```

```
6292 000552 017003 000000                      MOV     @(R0),R3        ;AND THE VALUE IS...
6293 000556 002415                             BLT     1$              ;TOO SMALL
6294 000560 022703 000014                      CMP     #MAXCHN,R3      ;STILL TOO BIG?
6295 000564 002412                             BLT     1$              ;YES
6296 000566 006303                             ASL     R3              ;Word offset
6297 000570 016302 000000G                     MOV     CHANNEL(R3),R2  ;Get Channel Pointer
6298 000574 005003                             CLR     R3              ;ASSUME FILE TYPE 0
6299 000576 013703 003112'                     MOV     DIMBLK,R3       ;GET FILE TYPE POINTER
6300 000602 001421                             BEQ     3$              ;NO, JUST EXIT
6301 000604 013700 003116'                     MOV     DIMTYP,R0       ;GET TYPE OF DIM ARG
6302 000610 001001                             BNE     2$              ;NOT LITERAL
6303 000612 104417                     1$:     DOMERR                  ;WRONGO!
6304 000614 022763 000001 000004      2$:     CMP     #1,A.ALEN(R3)   ;SCALAR?
6305 000622 001373                             BNE     1$              ;NOPE
6306 000624 011303                             MOV     (R3),R3         ;POINT TO DATA
6307 000626                                    CALL    @IRARGC(R0)     ;DO CONVERSION
6308 000632 013703 003222'                     MOV     RARGVA,R3       ;GET THE VALUE
6309 000636 002765                             BLT     1$              ;TOO SMALL
6310 000640 022703 000005                      CMP     #MAXTYP,R3      ;TOO BIG?
6311 000644 002762                             BLT     1$              ;YES
6312 000646 013704 003204'             3$:     MOV     RARGBL,R4       ;POINT TO RIGHT ARG
6313 000652 013701 003212'                     MOV     RARGTY,R1       ;AND RIGHT ARG TYPE
6314 000656 006303                             ASL     R3              ;Times two for lookup
6315 000660                                    RETURN
6316
6317 000000                                    .PSECT  TABLES
6318
6319 000000 000001 000001 000002 DATALN: .WORD  1,1,2,1,4,8.    ;LENGTH OF ITEMS IN FILE
     000006 000001 000004 000010
6320
6321 000014 000000 000000 177777 DATASH: .WORD  0,0,-1,0,-2,-3  ;Shift constant
     000022 000000 177776 177775
6322
6323 000030 000000               DATATY: .WORD  .CHARACTER       ;0 - INPUT LINE
6324 000032 000004                       .WORD  .INTEGER         ;1 - 1 byte NUMBER
6325 000034 000004                       .WORD  .INTEGER         ;2 - 2 byte NUMBER
6326 000036 000000                       .WORD  .CHARACTER       ;3 - APL CHARACTER
6327 000040 000006                       .WORD  .FLOAT           ;4 - SINGLE PRECISION FP
6328 000042 000006                       .WORD  .FLOAT           ;5 - DOUBLE PRECISION FP
6330 000044                              LIST
6331
6332
6333
6334
6335
6336 000044                      DOPINS  FPUT,152,T.DIM
6337                             .SBTTL  MIXED FUNCTION          FILE OUTPUT
6338
6339                             ;++
6340                             ; FUNCTIONAL DESCRIPTION:
6341                             ;       READS FILE SPECIFIED AND CONVERTS INFORMATION ACCORDING TO FILE TYPE.
6342                             ;       FORMAT SIMILAR TO FILE INPUT OPERATOR ABOVE.
6343                             ;--
6344
6345 000662                              CALL    FSETUP          ;SET UP FILE POINTERS
6346 000666                              PUSH    OCHAN           ;SAVE CURRENT OUTPUT CHANNEL
6347 000672 010237 005052'               MOV     R2,OCHAN        ;SET NEW OUTPUT CHANNEL
```

```
APLO05  MACRO M1113  08-FEB-80 09:29  PAGE 82-4
MIXED FUNCTION          FILE OUTPUT

 6348 000676  010502                          MOV     SI,R2               ;POINT TO LEFT ARG'S FRAME
 6349 000700  062702  177764                  ADD     #2*SI.LEN,R2        ;POINT TO TOP OF RIGHT ARG'S FRAME
 6350 000704  005737  003112'                 TST     DIMBLK              ;ANY DIM ARG?
 6351 000710  001402                          BEQ     1$                  ;NO
 6352 000712  062702  177764                  ADD     #2*SI.LEN,R2        ;YES, SKIP OVER IT
 6353 000716                          1$:     PUSH    <R0,R1,R2,R3>       ;SAVE REGISTERS
 6354 000726                                  CALL    SAME                ;GET NEW DATA IMAGE
 6355 000732                                  POP     <R3,R2,R1,R0>       ;RESTORE DATA
 6356 000742  026562  177776  177776          CMP     SI.VAL(SI),SI.VAL(R2) ;SAME STUFF?
 6357 000750  001002                          BNE     2$                  ;NO, GO...
 6358 000752  005062  177774                  CLR     SI.BITS(R2)         ;YES, ASSURE NON-IMMOLATION
 6359 000756  013702  005052'          2$:     MOV     OCHAN,R2            ;GET OUTPUT CHANNEL POINTER
 6360 000762  016300  000030'                 MOV     DATATY(R3),R0       ;GET RESULT TYPE
 6361 000766  001410                          BEQ     5$                  ;LITERAL DATA
 6363 000770  032762  020000  177756          BIT     #C.FASW,C.SBIT(R2);Was the file opened for Formatted ASCII?
 6364 000776  001003                          BNE     4$                  ;Yes, Can't do
 6366 001000  005701                          TST     R1                  ;IS RIGHT ARG NUMERIC?
 6367 001002  001004                          BNE     6$                  ;YES, TYPES MATCH!
 6368 001004  104417                  3$:     DOMERR                      ;Wrong data type
 6370 001006  104565                  4$:     ILLSW                       ;Wrong file type
 6372
 6373 001010  005701                  5$:     TST     R1                  ;IS RIGHT ARG LITERAL DATA?
 6374 001012  001374                          BNE     3$                  ;NO, WRONG FILE TYPE!
 6375
 6376 001014  006300                  6$:     ASL     R0                  ;RESULT TYPE
 6377 001016  006300                          ASL     R0                  ;  TIMES 4
 6378 001020  060001                          ADD     R0,R1               ;INDEX INTO LOOKUP
 6379 001022  016137  000000G 003220'         MOV     RARGCT(R1),RARGCV ;RIGHT ARG FETCH
 6380 001030  016401  000004                  MOV     A.ALEN(R4),R1       ;GET LENGTH OF ARG
 6381 001034  001534                          BEQ     14$                 ;NOTHING TO DO!
 6382 001036                                  PUSH    R3                  ;SAVE OFFSET
 6383 001040  011403                          MOV     (R4),R3             ;POINT TO FIRST VALUE TO WRITE
 6384 001042  062607                          ADD     (SP)+,PC            ;AND BRANCH TO FILE TYPE ROUTINE
 6385                                  ;
 6386                                  ;BRANCHES TO ROUTINES TO HANDLE DIFFERENT FILE TYPES
 6387                                  ;
 6388 001044  000405                          BR      7$                  ; 0 -- ASCII SEQUENTIAL
 6389 001046  000437                          BR      9$                  ; 1 -- BYTE VALUES
 6390 001050  000447                          BR      10$                 ; 2 -- INTEGER VALUES
 6391 001052  000462                          BR      11$                 ; 3 -- APL INTERNAL LITERAL FORMAT
 6392 001054  000467                          BR      12$                 ; 4 -- SINGLE PRECISION FLOATING POINT
 6393 001056  000503                          BR      13$                 ; 5 -- DOUBLE PRECISION FLOATING POINT
 6394
 6395                                  ;
 6396                                  ;ASCII SEQUENTIAL
 6397                                  ;
 6398
 6399 001060                          7$:
 6401 001060  032762  020000  177756          BIT     #C.FASW,C.SBIT(R2);Was the file opened as Formatted ASCII?
 6402 001066  001747                          BEQ     4$                  ;No, Can't write to it then
 6403 001070  005000                          CLR     R0                  ;For divide
 6404 001072  016401  000004                  MOV     A.ALEN(R4),R1       ;Get total array size
 6405 001076  116402  000002                  MOVB    A.DIMS(R4),R2       ;Get array rank
 6406 001102                                  CONINC  R2                  ;In case of a scalar
 6407 001106  006302                          ASL     R2                  ;Offset in bytes
 6408 001110  062702  000004                  ADD     #4,R2               ;Offset to dimension list
 6409 001114  060204                          ADD     R2,R4               ;Point to last dimension
```

```
APL005  MACRO M1113  08-FEB-80 09:29  PAGE 82-5
MIXED FUNCTION          FILE OUTPUT

 6410 001116  011404                          MOV     (R4),R4         ;Get the last dimension
 6411 001120                                  DIV     R4,R0           ;Get number of lines
 6412 001132  010401                  8$:     MOV     R4,R1           ;Copy the line length
 6413 001134                                  CALL    PUTFA           ;Write out a line
 6414 001140                                  SOB     R0,8$           ;Do all the lines
 6421 001144  000470                          BR      14$             ;AND EXIT
 6422                                 ;
 6423                                 ;BYTE VALUES
 6424                                 ;
 6425 001146                          9$:     CALL    @RARGCV         ;CALL FETCH
 6426 001152  013700  003222'                 MOV     RARGVA,R0       ;GET VALUE
 6427 001156                                  CALL    OCHAR           ;OUTPUT IT
 6428 001162                                  SOB     R1,9$           ;ALL ARGS
 6429 001166  000457                          BR      14$             ;EXIT
 6430                                 ;
 6431                                 ;INTEGER VALUES
 6432                                 ;
 6433 001170                          10$:    CALL    @RARGCV         ;CALL FETCH
 6434 001174  013737  003222' 005066'         MOV     RARGVA,OPTMP0   ;STORE INTEGER IN BUFFER
 6435 001202  012704  000002                  MOV     #2,R4           ;2 BYTES PER INTEGER
 6436 001206                                  CALL    PUTCH           ;WRITE 2 BYTES TO FILE
 6437 001212                                  SOB     R1,10$          ;LOOP FOR ARRAY
 6438 001216  000443                          BR      14$             ;EXIT
 6439                                 ;
 6440                                 ;APL INTERNAL LITERAL FORMAT
 6441                                 ;
 6442 001220  112300                  11$:    MOVB    (R3)+,R0        ;GET NEXT CHAR
 6443 001222                                  CALL    OCHAR           ;OUTPUT WITH NO TRANSLATIONS
 6444 001226                                  SOB     R1,11$          ;THAT'S EASY
 6445 001232  000435                          BR      14$             ;EXIT
 6446                                 ;
 6447                                 ;SINGLE PRECISION FLOATING POINT
 6448                                 ;
 6449 001234                          12$:    CALL    @RARGCV         ;FETCH VALUE
 6450 001240                                  STF     AC2,OPTMP0      ;STORE IN BUFFER
 6451 001250  012704  000004                  MOV     #4.,R4          ;4 BYTES PER SINGLE PRECISION NUMBER
 6452 001254                                  CALL    PUTCH           ;OUTPUT THEM
 6453 001260                                  SOB     R1,12$          ;LOOP
 6454 001264  000420                          BR      14$             ;EXIT
 6455                                 ;
 6456                                 ;DOUBLE PRECISION FLOATING POINT NUMBER
 6457                                 ;
 6458 001266                          13$:    CALL    @RARGCV         ;FETCH ARG
 6459 001272  005037  005072'                 CLR     OPTMP0+4        ;CLEAR TWO WORDS,
 6460 001276  005037  005074'                 CLR     OPTMP0+6        ;  JUST IN CASE
 6461 001302                                  STF     AC2,OPTMP0      ;STORE NUMBER IN BUFFER
 6462 001312  012704  000010                  MOV     #8.,R4          ;8 BYTES PER NUMBER
 6463 001316                                  CALL    PUTCH           ;OUTPUT THEM
 6464 001322                                  SOB     R1,13$          ;LOOP
 6465                                 ;
 6466                                 ;EXIT
 6467                                 ;
 6468 001326                          14$:    POP     OCHAN           ;RESTORE OUTPUT CHANNEL
 6469 001332                                  RETURN                  ;AND EXIT
 6470
 6471
 6472                                 ;
```

```
6473                                    ;SUBROUTINE TO MOVE CHARS FROM BYTE BUFFER TO FILE
6474                                    ;
6475
6476 001334                            PUTCH:  PUSH    R1              ;SAVE R1
6477 001336  012701  005066'                    MOV     #OPTMP0,R1      ;POINT TO BYTE BUFFER
6478 001342  112100                     1$:     MOVB    (R1)+,R0        ;GET NEXT CHAR
6479 001344                                     CALL    OCHAR           ;OUTPUT IT
6480 001350                                     SOB     R4,1$           ;LOOP
6481 001354                                     POP     R1              ;RESTORE R1
6482 001356                                     RETURN
6483
6485 001360                                     LIST
```

```
                                DOPINS  FPNT,154,T.DIM
                                .SBTTL  MIXED FUNCTION          SET FILE POINTER

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       The .PT function sets the file pointer on a channel.  The format
                                ;       is:
                                ;                               CHAN .PT [TYPE] PTR
                                ;
                                ;       Where CHAN is the channel number; [TYPE], which is optional, is
                                ;       the file type (taken as 0 if not present); and PTR is the pointer
                                ;       setting.  If PTR is non-negative then it is used to set the file
                                ;       pointer position in units of TYPE.  If PTR is negative then either
                                ;       the file length (TYPE = 0), or the pointer position (TYPE > 0) is
                                ;       returned.  The length is given in bytes, usually an even number of
                                ;       blocks, and the position is given in units of TYPE.  The units of
                                ;       a type zero (Formatted ASCII) file are records.
                                ;--

001360                                  CALL    FSETUP                  ;Set up file pointers
001364  022702  000256'                 CMP     #CHAN00,R2              ;Is it the terminal?
001370  001003                          BNE     1$                      ;No
001372  005000                          CLR     R0                      ;Yes, Ignore
001374  005001                          CLR     R1                      ;   the request
001376  000576                          BR      FPNT5                   ;And return origin as result

001400                          1$:
001400  005762  000070                  TST     F.BDB(R2)               ;Is the Channel Open?
001404  001001                          BNE     2$                      ;Yes
001406  104473                          NOTOPN                          ;Channel not open
001410  010337  005066'          2$:     MOV     R3,OPTMP0               ;Save the data type
001414  026427  000004  000001          CMP     A.ALEN(R4),#1           ;Length = 1 Element
001422  001401                          BEQ     3$                      ;Yes
001424  104412                          LENERR                          ;Length Error
001426  013700  003212'          3$:     MOV     RARGTY,R0               ;Right Arg Source Type
001432  011403                          MOV     (R4),R3                 ;Point to right argument
001434                                  CALL    @FRARGC(R0)             ;Get the conversion
001440  013703  005066'                 MOV     OPTMP0,R3               ;Restore Data Type
001444                                  LDF     AC2,AC0                 ;Copy to AC0
001452                                  CFCC                            ;Check the sign
001456                                  SUBF    FORIGN,AC0              ;Minus the origin
001466  002142                          BGE     FPNT5                   ;Set the pointer position
001470  016204  000014                  MOV     F.FFBY(R2),R4           ;Get the first free byte in the block
001474  032762  020000  177756          BIT     #C.FASW,C.SBIT(R2)      ;Is the file type zero?
001502  001004                          BNE     4$                      ;Yes
001504  016204  177762                  MOV     C.BUFL(R2),R4           ;Get buffer length
001510  166204  177766                  SUB     C.CCNT(R2),R4           ;Compute buffer pointer offset
001514  005703                  4$:     TST     R3                      ;Is the file type zero?
001516  001012                          BNE     FPNT1                   ;No, Return the currect pointer position
                                ;+
                                ; Find the file length
                                ;-
001520  016200  000010                  MOV     C.FLEN(R2),R0           ;Get length of the file
001524  016201  000012                  MOV     C.FLEN+2(R2),R1         ;Get both words for RSX
001530  032762  021000  177756          BIT     #C.CZ!C.FASW,C.SBIT(R2) ;Still in Write-only mode?
001536  001032                          BNE     FPNT2                   ;Yes
001540  005004                          CLR     R4                      ;No, Start at the beginning
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 83-1
MIXED FUNCTION          SET FILE POINTER

   6560 001542  000430                          BR      FPNT2                   ;Go get the difference between ends
   6561                                ;+
   6562                                ; Find the currect pointer position
   6563                                ;-
   6564 001544                         FPNT1:
   6566 001544  032762  020000  177756           BIT     #C.FASW,C.SBIT(R2)      ;A type zero file?
   6567 001552  001405                           BEQ     1$                      ;No
   6568 001554  016200  000034                   MOV     F.RCNM(R2),R0           ;Yes, Get the record number
   6569 001560  016201  000036                   MOV     F.RCNM+2(R2),R1         ;Low bits too
   6570 001564  000442                           BR      FPNT3                   ;And go
   6571 001566  016200  000064           1$:     MOV     C.BLOK(R2),R0           ;Get the current block number
   6572 001572  016201  000066                   MOV     C.BLOK+2(R2),R1         ;Low order bits too
   6577 001576  066304  000000'                  ADD     DATALN(R3),R4           ;Point to next avaliable element
   6578 001602  032762  002000  177756           BIT     #C.REWT,C.SBIT(R2)      ;Block written yet?
   6579 001610  001005                           BNE     FPNT2                   ;No
   6580 001612  005762  177760                   TST     C.BUFF(R2)              ;Is there a buffer?
   6581 001616  001002                           BNE     FPNT2                   ;Yes
   6582 001620  012704  000001                   MOV     #1,R4                   ;No, Then offset is at beginning of buffer
   6583
   6584 001624                         FPNT2:
   6586 001624  162701  000001                   SUB     #1,R1                   ;RSTS and RSX Files Start at Block 1!!!
   6587 001630  005600                           SBC     R0                      ;Double Precision
   6589 001632                                   ASHC    #9.,R0                  ;Times 512.
   6590 001646  060401                           ADD     R4,R1                   ;Add in offset from beginning of buffer
   6591 001650  005500                           ADC     R0                      ;All of it
   6592 001652  005703                           TST     R3                      ;Is it a type zero file?
   6593 001654  001414                           BEQ     FPNT4                   ;No
   6594 001656                                   ASHC    DATASH(R3),R0           ;Divide by the data type
   6595 001672  012703  000001           FPNT3:  MOV     #1,R3                   ;Prepare to negate the origin
   6596 001676  163703  000030'                  SUB     ORIGIN,R3               ;Negate the origin
   6597 001702  160301                           SUB     R3,R1                   ;Origin correction
   6598 001704  005600                           SBC     R0                      ;The whole thing
   6599 001706                         FPNT4:    PUSH    <R1,R0>                 ;Save Registers
   6600 001712                                   SETL                            ;Long Mode
   6601 001716                                   LDCLF   (SP)+,AC0               ;Convert to Floating Point
   6602 001724                                   SETI                            ;Integer Mode again
   6603 001730  012700  000010                   MOV     #A.DIM1+2,R0            ;Get the size of the result header
   6604 001734                                   PUSH    R0                      ;Save it for an offset
   6605 001736  063700  000062'                  ADD     NUMLEN,R0               ;Add in space for result value
   6606 001742                                   CALL    GETFRA                  ;Get result, SI frame for it
   6607 001746  062620                           ADD     (SP)+,(R0)+             ;Fill in array block header
   6608 001750  012720  103000                   MOV     #<A.ASB+.FLOAT>*400,(R0)+;  Type and Rank
   6609 001754  012720  000001                   MOV     #1,(R0)+                ;   Array length
   6610 001760  012720  000001                   MOV     #1,(R0)+                ;   First dimension
   6611 001764                                   STF     AC0,(R0)+               ;   Result value
Z       001770  010020                           MOV AC0,(R0)+
   6612 001772                                   RETURN                          ;And exit
   6613                                ;+
   6614                                ; Set the file pointer position
   6615                                ;-
   6616 001774                         FPNT5:    CFCC                            ;Is the new pointer negative?
   6617 002000  100405                           BMI     1$                      ;Yes, Domain Error
   6621 002002  032762  020000  177756           BIT     #C.FASW,C.SBIT(R2)      ;File opened as Formatted ASCII?
   6622 002010  001402                           BEQ     2$                      ;Yes, OK
   6623 002012  104565                           ILLSW                           ;No, No random access on ASCII files
   6625 002014  104417                   1$:     DOMERR                          ;Yes, Can't set ptr in a seq file
   6626 002016  042762  001000  177756   2$:     BIC     #C.CZ,C.SBIT(R2)        ;Direct access, Don't write CNTL-Z
```

```
6628 002024                                    PUSH    C.BLOK+2(R2)            ;Get low word of Block number
6632                                    ;+
6633                                    ; Determine the block number
6634                                    ;-
6635 002030                                    LDCIF   DATALN(R3),AC1          ;Convert data type size to floating
6636 002040                                    MULF    AC1,AC0                 ;Convert pointer value to bytes
6637 002046                                    SETL                            ;Set to Long Integer
6638 002052                                    STCFL   AC0,-(SP)               ;Convert to Long Integer form
6639 002060                                    SETI                            ;Back to short integer
6640 002064                                    POP     <R0,R1>                 ;Restore Registers
6641 002070  010103                            MOV     R1,R3                   ;Get low word
6642 002072  042703  177000                    BIC     #^C<OBUFLN-1>,R3        ;Position in buffer
6643 002076                                    ASHC    #-9.,R0                 ;Convert it to a block count
6645 002112  005201                            INC     R1                      ;Bump by 1 If RSTS or RSX
6647 002114                                    PUSH    R1                      ;Save block number
6648 002116  166601  000002                    SUB     2(SP),R1                ;Minus the current position
6649 002122  103416                            BLO     3$                      ;Behind, We must read in a new block
6650 002124  005762  177760                    TST     C.BUFF(R2)              ;Is a buffer allocated?
6651 002130  001413                            BEQ     3$                      ;No, So we must read from the file
6652 002132                                    ASH     #9.,R1                  ;Convert the distance away to bytes
6653 002146  103404                            BCS     3$                      ;It is more than a block away
6654 002150  060301                            ADD     R3,R1                   ;Add in buffer offset
6655 002152  020162  177762                    CMP     R1,C.BUFL(R2)           ;Are we still in the current block?
6656 002156  103457                            BLO     6$                      ;Yes
6657                                    ;+
6658                                    ; Write out current buffer if necessary, and read in a new block
6659                                    ;-
6660 002160  032762  004000  177756    3$:     BIT     #C.FLSH,C.SBIT(R2)      ;Has any data in the buffer changed?
6661 002166  001402                            BEQ     4$                      ;No, Then we don't have to rewrite the block
6662 002170                                    CALL    WRITEB                  ;Write out the current buffer
6663 002174  016201  177762            4$:     MOV     C.BUFL(R2),R1           ;Get the buffer size
6664 002200  062701  000777                    ADD     #<OBUFLN-1>,R1          ;This does nothing but
6665 002204  042701  000777                    BIC     #<OBUFLN-1>,R1          ;  set R1
6666 002210                                    ASH     #-9.,R1                 ;  to 1
6667 002224  011600                            MOV     (SP),R0                 ;Get the new block pointer
6669 002226  020062  000012                    CMP     R0,C.FLEN+2(R2)         ;Are we extending the file?
6673 002232  103420                            BLO     5$                      ;No, So the block exist
6675 002234  011662  000066                    MOV     (SP),C.BLOK+2(R2)       ;Save the block pointer
6679 002240  052762  002000  177756          BIS     #C.REWT,C.SBIT(R2)      ;Mark that the new block doesn't exist
6680 002246  010301                            MOV     R3,R1                   ;Copy the buffer offset
6681 002250  005762  177760                    TST     C.BUFF(R2)              ;Is there a buffer?
6682 002254  001020                            BNE     6$                      ;Yes
6683 002256  016200  177762                    MOV     C.BUFL(R2),R0           ;No, Get the buffer size
6684 002262                                    CALL    GETCOR                  ;Allocate a buffer
6685 002266  010062  177760                    MOV     R0,C.BUFF(R2)           ;Set the buffer pointer
6686 002272  000411                            BR      6$                      ;Skip reading a non-existant block
6687
6688 002274                            5$:
6690 002274  011662  000066                    MOV     (SP),C.BLOK+2(R2)       ;Save the block to read
6694 002300                                    CALL    READB                   ;Read the block into the buffer
6696 002304  160162  000066                    SUB     R1,F.BKVB+2(R2)         ;Back up by the
6697 002310  005662  000064                    SBC     F.BKVB(R2)              ;   size of the buffer
6701 002314  010301                            MOV     R3,R1                   ;Copy the offset into the buffer
6702                                    ;+
6703                                    ; Reset pointer into buffer
6704                                    ;-
6705 002316  016262  177762  177766    6$:     MOV     C.BUFL(R2),C.CCNT(R2)   ;Assume the buffer is full
```

MIXED FUNCTION SET FILE POINTER

```
 6706 002324  160162  177766          SUB     R1,C.CCNT(R2)           ;Minus the space remaining in the buffer
 6707 002330  066201  177760          ADD     C.BUFF(R2),R1           ;Convert into pointer into the buffer
 6708 002334  010162  177764          MOV     R1,C.CPTR(R2)           ;And change the file pointer
 6709 002340  062706  000004          ADD     #4,SP                   ;Pop junk
 6710 002344  010304                  MOV     R3,R4                   ;Copy the buffer offset
 6711 002346  013703  005066'         MOV     OPTMP0,R3               ;Get back the data type
 6712 002352  000137  001544'         JMP     FPNT1                   ;And return the new pointer position
 6714 002356                          LIST
```

L 11

```
APLO05  MACRO M1113  08-FEB-80 09:29  PAGE 84
MIXED FUNCTION          SET FILE POINTER

   6716 002356                          MOPINS  1SCN,130
   6723 002356                                  LIST
   6724
   6725
   6726 002356                          MOPINS  SCAN,62,T.DIM
   6732 002356                                  LIST
   6733
   6734
   6735 002356                          MOPINS  1RDT,124
   6741 002356                                  LIST
   6742
   6743
   6744 002356                          MOPINS  RDCT,60,T.DIM
   7044 002356                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 85
MIXED FUNCTION          SET FILE POINTER

   7046 002356                          MOPINS  GETIDN,0
   7193 002356                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 86
MIXED FUNCTION          SET FILE POINTER

   7195 002356                          MOPINS  SATAN,0
   7404 002356                                  LIST
```

MIXED FUNCTION   SET FILE POINTER

```
   7406 002356                          MOPINS  OUTR,.56
   7476 002356                                  LIST
```

```
   7478 002356                          MOPINS  DOUTER,0
   7504 002356                                  LIST
   7505
   7506
   7507
   7508
   7509
   7510 002356                          MOPINS  PRDSET,0
   7564 002356                                  LIST
   7565
   7566
   7567
   7568
   7569
   7570 002356                          MOPINS  INSPAC,0
   7612 002356                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 89
MIXED FUNCTION          SET FILE POINTER

   7614 002356                          DOPINS  INNR,156
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 90
MIXED FUNCTION          SET FILE POINTER

   7960 002356                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 91
MIXED FUNCTION          SET FILE POINTER

   7962 002356                          MOPINS  SUBR,0
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 92
MIXED FUNCTION          SET FILE POINTER

   8725 002356                                  LIST
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 93
MIXED FUNCTION          SET FILE POINTER

   8727 002356                          MOPINS  RAND,0
   8771 002356                                  LIST
```

```
APLO05  MACRO M1113  08-FEB-80 09:29  PAGE 94
MIXED FUNCTION          SET FILE POINTER

     1          000001                              .END
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 94-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.BLOK= 000064  D.CIRC= 000001  D.1EXP= 000002
AC1    =%000001          CHRLBR= 000073  C.BLSZ= 000140  D.COMB= 000001  D.1ROT= 000003
AC2    =%000002          CHRLPR= 000072  C.BUFF= 177760  D.COMP= 000001  EDICHR  005063R  025
AC3    =%000003          CHRMNS= 000137  C.BUFL= 177762  D.CONT= 000006  EDICNT  005064R  025
AC4    =%000004          CHRMUL= 000075  C.CCNT= 177766  D.CTLM= 000000  EDIFLG  005062R  025
AC5    =%000005          CHRNEG= 000100  C.CPTR= 177764  D.DEAL= 000003  EDIT   = 000003
ALLINT  003274R     025  CHRPDL= 000271  C.CZ  = 001000  D.DECD= 000002  EIS    = 000000
ARYRET  003276R     025  CHRPLU= 000055  C.DIG = 000040  D.DIV = 000000  ELESIZ  005132R  025
ASCQUO= 000042           CHRPWR= 000120  C.FASW= 020000  D.DMD = 000010  ENDCOR  000004R  025
A.ALEN= 000004           CHRQST= 000121  C.FLEN= 000010  D.DQAD= 177777  EOF    = 104475
A.ARG = 000040           CHRQUO= 000113  C.FLSH= 004000  D.DROP= 000002  ERRMAX= 000166
A.ASB = 000200           CHRRBR= 000047  C.FRC = 000001  D.ENCD= 000002  ERRNUM  004772R  025
A.BITS= 000003           CHRRPR= 000042  C.IDEN= 000010  D.EQ  = 000000  ERRPTR  004770R  025
A.CON = 000020           CHRSEM= 000074  C.NAME= 000102  D.EXPD= 000002  ERR0    004774R  025
A.DIMS= 000002           CHRSMO= 000112  C.NOBF= 000400  D.FGET= 000005  EXEBUF  005054R  025
A.DIM1= 000006           CHRSPA= 000040  C.NUM = 000020  D.FMT = 000004  E$CT    005056R  025
A.NUM = 000006           CHRUPA= 000131  C.OP  = 000004  D.FPNT= 000005  E$PT    005060R  025
A.PTR = 000000           CHR.BS= 000010  C.OPN = 010000  D.FPUT= 000005  FA.APD= 000100
A.TMP = 000010           CHR.CC= 000003  C.RDLN= 177776  D.GE  = 000000  FA.CRE= 000010
BADDIR= 104463           CHR.CR= 000015  C.REWT= 002000  D.GT  = 000000  FA.DLK= 001000
BADERR= 104437           CHR.CZ= 000032  C.RND = 000010  D.IBM = 000002  FA.ENB= 100000
BLZBUB  005116R     025  CHR.GT= 000046  C.SBIT= 177756  D.IND = 000002  FA.EXC= 002000
BRSELF  005046R     025  CHR.LF= 000012  C.SPFX= 000100  D.INNR= 000013  FA.EXT= 000004
CCNT    005072R     025  CHR.LT= 000043  C.SPGL= 000200  D.INTR= 000006  FA.NSP= 000100
CHANEX  003060R     025  CHR.SI= 000017  C.SPLX= 000001  D.LE  = 000000  FA.NSR= 000200
CHANNE= ****** GX        CHR.SL= 000057  C.SPRS= 000002  D.LOG = 000001  FA.POS= 010000
CHAN00  000256R     025  CHR.SO= 000016  C.SRC = 177774  D.LT  = 000000  FA.RD  = 000001
CHAN01  000330R     025  CHR.TB= 000011  C.SRCP= 000024  D.MAX = 000000  FA.RWD= 004000
CHAN02  000512R     025  CHR.UL= 000106  C.TT  = 000004  D.MEM = 000002  FA.SEQ= 040000
CHAN03  000674R     025  CHR..A= 000141  C.WIDC= 177770  D.MIN = 000000  FA.SHR= 000040
CHAN04  001056R     025  CHR..E= 000145  C.WIDR= 177772  D.MUL = 000000  FA.TMP= 000020
CHAN05  001240R     025  CHR..S= 000163  DATALN  000000R  002 D.NE  = 000000  FA.WCK= 020000
CHAN06  001422R     025  CHR..T= 000164  DATASH  000014R  002 D.NN  = 000000  FA.WRT= 000002
CHAN07  001604R     025  CHR..U= 000165  DATATY  000030R  002 D.NR  = 000000  FD.BLK= 000010
CHAN08  001766R     025  CHR..Z= 000172  DEBUG = 000000      D.OR  = 000000  FD.CCL= 000002
CHAN09  002150R     025  CHR..0= 000060  DEFERR= 104405      D.PLUS= 000000  FD.COM= 020000
CHAN10  002332R     025  CHR..1= 000061  DEVIL   005114R  025 D.PSUB= 000006  FD.CR  = 000002
CHAN11  002514R     025  CH.AND= 000001  DEVNTF= 104470      D.PWR = 000001  FD.DIR= 000010
CHAN12  002676R     025  CMDBUF  003250R  025 DIGITS  000006R  025 D.QQAD= 000000  FD.FTN= 000001
CHAN13  000022R     023  CMDMAX= 000010      DIMAX = 000020      D.QUAD= 000000  FD.F11= 040000
CHAN14  000204R     023  CNTSIZ  005130R  025 DIMBLK  003112R  025 D.RES = 000000  FD.INS= 000010
CHNOPN= 104471           COMBIN  005076R  025 DIMPTR  003114R  025 D.RHO = 000000  FD.ISP= 002000
CHRAND= 000051           COMCNT  005126R  025 DIMTYP  003116R  025 D.ROT = 000003  FD.MNT= 100000
CHRBEL= 000007           COMERR= 104425      DIRFUL= 104466      D.SCON= 000006  FD.OSP= 004000
CHRBKR= 000133           COMPTR  005122R  025 DOMERR= 104417      D.SDIF= 000006  FD.PLC= 000004
CHRBOL= 000200           COMRTN  003156R  025 DPTERR= 104416      D.SUB = 000000  FD.PRN= 000004
CHRBOX= 000114           COMSKP  005124R  025 DYDCNT  003224R  025 D.SUBS= 000006  FD.PSE= 010000
CHRCMA= 000054           CORPTR  005120R  025 D$FGET  000000RG  020 D.TAKE= 000002  FD.RAH= 000001
CHRCOL= 000076           CS.ASS= 000200      D$FPNT  001360RG  020 D.TRAN= 000004  FD.RAN= 000002
CHRDEL= 000107           CS.BOO= 000010      D$FPUT  000662RG  020 D.UNON= 000006  FD.REC= 000001
CHRDIV= 000053           CS.FLO= 000020      D.AND = 000000      D.$MAX= 000000  FD.RWM= 000001
CHRDLR= 000176           CS.IDN= 000040      D.BXCT= 000007      D.$MIN= 000000  FD.SDI= 000020
CHRDLT= 000110           CS.ONE= 000100      D.BXIO= 000007      D.$MUL= 000000  FD.SQD= 000040
CHRDOT= 000056           CTEPS   000152R  025 D.BXNL= 000007      D.$PLU= 000000  FD.TTY= 000004
CHREP = 000105           CTMAX   000212R  025 D.BXPP= 000007      D.$RES= 000000  FD.WBH= 000002
CHRERR= 104431           CTRMAX  000172R  025 D.BXPW= 000007      D.$SUB= 000000  FF.CHR= 000005
CHRFIL= 000177           C.ALOC= 000162      D.BXRL= 000007      D.1COM= 000001  FF.NV = 000003
```

APLOO5 MACRO M1113 08-FEB-80 09:29 PAGE 94-2
SYMBOL TABLE

```
FF.POE= 000002          F.CNTG= 000034          F.STOP= 000004          LEXPTR  003100R 025 M.GDWN= 000003
FF.RWD= 000001          F.CODE= 000010          F.STP = 000001          LFTOP   005066R 025 M.GETI= 000006
FF.RWF= 000006          F.DFNB= 000046          F.SUSP= 000002          LINC    005102R 025 M.IBM = 000003
FF.SPC= 000004          F.DSPT= 000044          F.TRC = 000002          LINE    005016R 025 M.IDN = 000000
FGET1   000260R     020 F.DVNM= 000134          F.UNIT= 000136          LININC  005026R 025 M.INSP= 000006
FHEADR  005012R     025 F.EFBK= 000010          F.URBD= 000020          LISIZE= 000020      M.IOTA= 000000
FHRET   005040R     025 F.EFN = 000050          F.VAL = 177776          LITOP   005072R 025 M.LIT = 000205
FILNTF= 104467          F.END = 002000          F.VBN = 000064          LNLNG = 104541      M.LN  = 000001
FIS   = 000000          F.EOBB= 000032          F.VBSZ= 000060          LOCLCT  005036R 025 M.MMD = 000010
FISCON= 046000          F.ERR = 000052          GETBYT= ****** GX       LOWCOR  000002R 025 M.NEG = 000000
FIXLIN= ****** GX       F.FACC= 000043          GETCOR= ****** GX       LSTERP  004766R 025 M.NOT = 000000
FIXPTR  003074R     025 F.FFBY= 000014          GETFA = ****** GX       LWIDTH  000070R 025 M.NUB = 000006
FLAGS   004764R     025 F.FLAG= 000002          GETFRA= ****** GX       L.BITS= 000003      M.NUM = 000204
FLCBUF  003250R     025 F.FLB = 000100          GETSCA= ****** GX       L.CNT = 000002      M.OUTR= 000012
FLINE   005014R     025 F.FNAM= 000110          GNOME   003102R 025 L.PRNT= 000001      M.PITM= 000000
FLOEXP  003270R     025 F.FNB = 000102          HASHC   005136R 025 L.PTR = 000000      M.PRDS= 000006
FLONEG  003272R     025 F.FNDF= 000010          HASHTB  005140R 025 L.PTR1= 000004      M.QQAD= 000000
FLSWTB  000222R     025 F.FTYP= 000116          HSHMSK= 177601      L.SUBS= 000002      M.QUAD= 000000
FLTLEN= 000010          F.FUN = 000010          HSHSIZ= 000200      MACHIN= 000106      M.RAND= 000012
FNINSI  005044R     025 F.FVER= 000120          IASH  = ****** GX       MAXCHN= 000014      M.RAVL= 000000
FORIGN  000032R     025 F.HEAD= 000006          IASHC = ****** GX       MAXTYP= 000005      M.RCIP= 000000
FO.APD= 000106          F.HIBK= 000004          ICHAN   005050R 025 MONBLK  003120R 025 M.RDCT= 000011
FO.MFY= 000002          F.LBL = 000020          IDIV  = ****** GX       MONCVT  003132R 025 M.REVS= 000003
FO.RD = 000001          F.LINE= 000012          ILFNAM= 104464          MONLEN  003130R 025 M.ROLL= 000000
FO.UPD= 000006          F.LOC = 000004          ILLCHN= 104540          MONPTR  003122R 025 M.SATA= 000001
FO.WRT= 000016          F.LSTC= 000004          ILLSW = 104565          MONSFT  003126R 025 M.SCAN= 000011
FPMODE  000042R     025 F.LUN = 000042          INDERR= 104410          MONTYP  003124R 025 M.SHAP= 000000
FPNT1   001544R     020 F.MBCT= 000054          INNTMP  003250R 025 MONVAL  003134R 025 M.SIGN= 000000
FPNT2   001624R     020 F.MBC1= 000055          INNTM2  005016R 025 M.ABS = 000000      M.SUBR= 000001
FPNT3   001672R     020 F.MBFG= 000056          INPTR   000064R 025 M.BOL = 000200      M.SYM = 000203
FPNT4   001706R     020 F.MON = 000002          INSNAM= 104462      M.BXAV= 000007      M.SYN = 000201
FPNT5   001774R     020 F.NIL = 000001          IOERR = 104477      M.BXCR= 000007      M.TRAN= 000012
FPP   = 000000          F.NRBD= 000024          IRARGC= ****** GX       M.BXCT= 000007      M.$ABS= 000000
FPPLEN= 000004          F.NREC= 000030          I.CNT = 177776          M.BXEX= 000007      M.$CEI= 000000
FRARGC= ****** GX       F.OPN = 000010          I.DIM = 177772          M.BXFX= 000007      M.$FLR= 000000
FREBLK  007400R     025 F.OVBS= 000030          I.LEN = 000012          M.BXIO= 000007      M.$IDN= 000000
FSETUP  000542R     020 F.PREV= 000000          I.OERR= 000062          M.BXLC= 000003      M.$NEG= 000000
FUZZ    000142R     025 F.PRNT= 100000          I.OLD = 177770          M.BXNC= 000007      M.$SIG= 000000
FXPTR1  003076R     025 F.PROT= 010000          I.PTR = 177774          M.BXNL= 000007      M.1RDT= 000011
F$EVAL= 000100          F.RACC= 000016          I.STR = 177766          M.BXPP= 000007      M.1REV= 000003
F$EVI = 000200          F.RATT= 000001          LABERR= 104406          M.BXPW= 000007      M.1SCN= 000011
F$EXEC= 000400          F.RCNM= 000034          LABLPT  005042R 025 M.BXRL= 000007      NAMBUF  003240R     025
F$FUNC= 001000          F.RCTL= 000017          LARGBL  003162R 025 M.BXTS= 000007      NB.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LARGCV  003176R 025 M.BXWA= 000007      NB.DIR= 000100
F.ALOC= 000040          F.REDF= 000040          LARGFL  003160R 025 M.CEI = 177777      NB.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LARGLE  003174R 025 M.CHR = 000202      NB.SD1= 000400
F.BDB = 000070          F.RET = 177772          LARGLN  003166R 025 M.CMT = 000206      NB.SD2= 001000
F.BGBC= 000057          F.RSIZ= 000002          LARGPT  003164R 025 M.DOUT= 000006      NB.SNM= 000040
F.BITS= 177774          F.RTYP= 000000          LARGSF  003172R 025 M.DQAD= 000006      NB.STP= 000020
F.BKDN= 000026          F.R2  = 000014          LARGTY  003170R 025 M.ENCD= 000004      NB.SVR= 000010
F.BKDS= 000020          F.R3  = 000016          LARGVA  003200R 025 M.EXE = 000000      NB.TYP= 000002
F.BKEF= 000050          F.R4  = 000020          LBINC   005100R 025 M.EXP = 000001      NB.VER= 000001
F.BKP1= 000051          F.SEQN= 000100          LBUFF   003316R 025 M.FACT= 000001      ND.DEV= 000200
F.BKST= 000024          F.SPDV= 000072          LBUFLN= 000621          M.FIL = 000177      ND.EXT= 000002
F.BKVB= 000064          F.SPUN= 000074          LCOUNT  005106R 025 M.FLR = 177777      ND.NAM= 000004
F.BR  = 000020          F.STBK= 000036          LENERR= 104412          M.FMT = 000004      ND.PPN= 000100
F.CHR = 000075          F.STK = 000022          LENTAB  000054R 025 M.GDUP= 000003      ND.$EX= 000000
```

APLOO5 MACRO M1113 08-FEB-80 09:29 PAGE 94-3
SYMBOL TABLE

```
ND.$E$= 000020          OPTMP3  005074R     025 ROTINC  005070R     025 SYNERR= 104407          T.NUM = 002000
ND.$NM= 000000          OPTMP4  005076R     025 ROUND   000010R     025 SYSERR= 104402          T.OPFN= 000001
ND.$NS= 000040          OPTMP5  005100R     025 ROVER   003312R     025 SYSTAT  000000R     024 T.OTHR= 040000
ND.$PN= 001400          OPTMP6  005102R     025 RSTS  = 000000          SY.BAD= 000001          T.SOP = 000002
NEGSGN  005063R     025 OPTMP7  005104R     025 RSX   = 000115          SY.BRK= 100000          T.SPOP= 000040
NINPT1  003302R     025 OPTMP8  005106R     025 RTOP    005070R     025 SY.BS = 000010          T.TMP = 000200
NINPT2  003300R     025 OPTMP9  005110R     025 RT11  = 000000          SY.CLR= 000040          T.VAL = 100000
NINSTK  003304R     025 OPTM10  005112R     025 R.BITS= 000003          SY.ERP= 000020          T.VAR = 001000
NINTMP  003310R     025 OPTM14  005026R     025 R.FIX = 000001          SY.NOU= 000100          UPPER = 177400
NINTYP  003306R     025 ORIGIN  000030R     025 R.LCT = 000002          SY.OPN= 000002          UTNAME  000344R     023
NOTAVL= 104472          OUTPTR  000066R     025 R.LN  = 000002          SY.PMT= 001000          VALERR= 104413
NOTOPN= 104473          OVROPR= 000005          R.LNUM= 000010          SY.SRT= 000400          VARBLE  007400R     025
NPRT1   004762R     025 PATCH   007344R     025 R.NEXT= 000006          SY.SVC= 000004          VARBLS  000000R     025
NULL    007364R     025 PRECSN  000052R     025 R.OFF = 000000          SY.SVL= 000200          VERNUM  000000R     025
NUMCHN= 000020          PUTCH   001334R     020 R.PREV= 000004          S.AB  = 177776          VRSION= 000002
NUMDFL  003230R     025 PUTFA = ****** GX       R.PTR = 000000          S.ASB = 100000          WIDTH   000072R     025
NUMEFL  003233R     025 RANDAC  003064R     025 R.SEQ = 000003          S.BFHD= 000020          WRITEB= ****** GX
NUMESN  003236R     025 RANDY   003062R     025 R.VAR = 000002          S.BITS= 000002          WROUND  000020R     025
NUMFLG  003230R     025 RANERR= 104411          R6    =%000006          S.CHR1= 000003          WSDATE  000136R     025
NUMLEN  000062R     025 RARGBL  003204R     025 R7    =%000007          S.FATT= 000016          WSFULL= 104403
NUMNEG  003232R     025 RARGCT= ****** GX       SAME  = ****** GX       S.FDB = 000140          WSNAME  000074R     025
NUMQFL  003234R     025 RARGCV  003220R     025 SI    =%000005          S.FLB = 040000          WSTIME  000132R     025
NUMRIG  003231R     025 RARGFL  003202R     025 SIDAMN= 104433          S.FNAM= 000006          $CFCC = ***** GX
NUMWID  003226R     025 RARGLE  003216R     025 SIFRAM  003314R     025 S.FNB = 000036          $FIS  = ****** G
NUMZFL  003235R     025 RARGLN  003210R     025 SISTAK  006350R     025 S.FNBW= 000017          $FPP  = ****** G
NYIERR= 104432          RARGPT  003206R     025 SITOP   006346R     025 S.FNTY= 000004          $FPPLN= ****** G
N.BLSZ= 000036          RARGSF  003214R     025 SITP    007144R     025 S.FTYP= 000002          $LDCIF= ****** GX
N.DESC= 000020          RARGTY  003212R     025 SIWARN= 000200          S.LEN = 000002          $LDCLF= ****** GX
N.DEV = 000032          RARGVA  003222R     025 SIZTAB  000044R     025 S.LINK= 177774          $LDF  = ****** GX
N.DEVN= 000034          RBASE   005074R     025 SI.BIT= 177774          S.NFEN= 000020          $MACHN= ****** G
N.DID = 000024          RCOUNR  005112R     025 SI.LEN= 177772          S.TYPE= 140000          $MULF = ****** GX
N.DVNM= 000032          RCOUNT  005110R     025 SI.RET= 177772          S.VAL = 000000          $RSTS = ****** G
N.EXT = 000014          READB = ****** GX       SI.VAL= 177776          TKDIMS  005072R     025 $RSX  = ****** G
N.FID = 000000          REDCNT  005070R     025 SKPSIZ  005134R     025 TKFRAM  005066R     025 $RT11 = ****** G
N.FNAM= 000006          REDIGN  005066R     025 SPHOLD  000004R     024 TKLEFT  005070R     025 $SETI = ****** GX
N.FTYP= 000014          REDOFF  005072R     025 SPINIT  000002R     024 TMODE   000403R     023 $SETL = ****** GX
N.FVER= 000016          REENTR  003060R     025 STKINC= 000774          TMPTAG= 000166          $STCFL= ****** GX
N.NAME= 000006          REGTYP= 000001          STOP  = 104420          TTYBUF  004140R     025 $STF  = ****** GX
N.NEXT= 000022          RESBIT  003142R     025 SUBCNT  003104R     025 TTYLEN= 000622          $SUBF = ****** GX
N.PPN = 000024          RESCVT  003152R     025 SUBDCT  003106R     025 TURKEY= 104436          .BOOLE= 000002
N.STAT= 000020          RESERR= 104440          SUBPTR  003110R     025 TYPE  = 000166          .CHARA= 000000
N.UNIT= 000034          RESLEN  003150R     025 SYMB1   005344R     025 T.BOL = 020000          .DEF. = 000002
OBUFLN= 001000          RESPTR  003140R     025 SYMF    005340R     025 T.CON = 004000          .FLOAT= 000006
OCHAN   005052R     025 RESSFT  003146R     025 SYMPT1  005346R     025 T.DIM = 000100          .FSRCB= ****** G
OCHAR = ****** GX       RESTYP  003144R     025 SYMTAB  005352R     025 T.FUN = 010000          .INTEG= 000004
OMODE   000402R     023 RESULT  003136R     025 SYMTBF= 104404          T.LIST= 000400          .MOLUN  000404R     023
OPTMP0  005066R     025 RESVAL  003154R     025 SYMTOP  005350R     025 T.LIT = 000020          ...GBL= 000000
OPTMP1  005070R     025 RINC    005104R     025 SYMTP   006272R     025 T.MON = 000010          ...TPC= 000020
OPTMP2  005072R     025 RLTBAS  005066R     025 SYMWRN= 000054          T.NIL = 000004

. ABS.  000000      000
        000000      001
TABLES  000044      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
```

```
APLOO5  MACRO M1113  08-FEB-80 09:29  PAGE 94-4
SYMBOL TABLE

DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    002356      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  13594 WORDS  ( 54 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:39
APLOO5,DM1:[305,301]APLOO5/-SP=APLRX1,APLDEF,APLOO5,APLOP,END
```

FILEID**APL006

```
  AAAAAA      PPPPPPPP      LL              000000        000000        666666
  AAAAAA      PPPPPPPP      LL              000000        000000        666666
AA      AA    PP      PP    LL            00    00    00    00    66
AA      AA    PP      PP    LL            00    00    00    00    66
AA      AA    PP      PP    LL            00    00    00  0000    66
AA      AA    PP      PP    LL            00    00    00  0000    66
AA      AA    PPPPPPPP      LL            00    00    00  00  00  66666666
AA      AA    PPPPPPPP      LL            00    00    00  00  00  66666666
AAAAAAAAAA    PP            LL            00    00    0000    00  66      66
AAAAAAAAAA    PP            LL            00    00    0000    00  66      66
AA      AA    PP            LL            00    00    00      00  66      66    ....
AA      AA    PP            LL            00    00    00      00  66      66    ....
AA      AA    PP            LLLLLLLLLL      000000        000000      666666    ....
AA      AA    PP            LLLLLLLLLL      000000        000000      666666    ....


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 09:33
TABLE OF CONTENTS


```
     2-    1     APLDEF
     2-    8     A P L
     2-    9
     2-   10     A PROGRAMMING LANGUAGE
     2-   11
     2-   12     STEVE POULSEN & DAVE VANN
     2-   13     11-Jul-76
     2-   14     ROBERT GOTTLIEB & PETER HAYNES
     2-   15     2-Feb-79
     2-   16
     2-   17     V2
     2-   18
     2-   19
     2-   20     COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25     COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26     COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27     BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28     USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29     TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30     DEC.
     2-   31
     2-   32     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33     AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34     CORPORATION.
     2-   35
     2-   36     DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37     SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46     DEFINITIONS     ASSEMBLY PARAMETERS
     4-  139     DEFINITIONS     OPERATOR OVERLAYS
     5-  268     DEFINITIONS     REGISTERS
     6-  331     DEFINITIONS     MEMORY ORGANIZATION
     7-  377     DEFINITIONS     ORDERING PSECTS
     8-  426     DEFINITIONS     SI PARSE FRAMES
     9-  474     DEFINITIONS     TOKEN DESCRIPTION BITS
    10-  515     DEFINITIONS     SI EXECUTION FRAMES
    11-  580     DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-  607     DEFINITIONS     CHRTAB BITS
    12-  624     DEFINITIONS     SYSTEM STATUS BITS
    13-  640     DEFINITIONS     F-FRAMES
    14-  724     DEFINITIONS     FUNCTION LINE BLOCKS
    16-  807     DEFINITIONS     ARRAY STORAGE BLOCK
    17-  869     DEFINITIONS     LIST STORAGE BLOCK
    18-  911     DEFINITIONS     INDEXING STACK FRAME
    19-  940     DEFINITIONS     SYMBOL TABLE ENTRY
    20-  969     DEFINITIONS     CHANNEL BLOCK
    22- 1085     DEFINITIONS             RSX DATA
    23- 1110     DEFINITIONS     NAME BLOCKS
    33- 1612     DEFINITIONS     MACROS
    35- 1905     DEFINITIONS     APL CHARACTERS
    36- 1967     DEFINITIONS     ERRORS
    68- 3924     MIXED FUNCTION          NUB - MONADIC UP U - ELIMINATE DUPLICATES
```

APL006 MACRO M1113 08-FEB-80 09:33
TABLE OF CONTENTS

68- 3993	MIXED FUNCTION	UNION
68- 4029	MIXED FUNCTION	INTERSECTION
68- 4049	MIXED FUNCTION	SET DIFFERENCE
69- 4164	MIXED FUNCTION	LEFT U - STRICT CONTAINMENT
69- 4183	MIXED FUNCTION	UNDERSCORE LEFT U - CONTAINMENT
69- 4204	MIXED FUNCTION	RIGHT U - PROPER SUBSET OF
69- 4213	MIXED FUNCTION	UNDERSCORE RIGHT U - SUBSET OF
75- 5159	MIXED FUNCTION	DEL QUAD INPUT
85- 7047	GETIDN	CREATE IDENTITY ELEMENT ARRAY
88- 7479	DOUTER	PERFORMS THE ACTUAL OUTER PRODUCT OPERATION
88- 7511	PRDSET	SETUP ROUTINE FOR INNER AND OUTER PRODUCTS
88- 7571	INSPAC	ALLOCATE SPACE FOR INNER PRODUCT

```
                                RT11    =  0            ;NOT RT-11
                                RSTS    =  0            ;NOT RSTS/E
                                RSX     = 'M            ;RSX-11/M

                                EIS     =  0            ;NO EIS
                                FIS     =  0            ;NO FIS
                                FPP     =  0            ;NO FPP

                                FPPLEN  =  4.           ;SINGLE PRECISION

                                DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 09:33  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLOO6  MACRO M1113  08-FEB-80 09:33  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APLOO6
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO6 IS:
      9                                 ;
     10                                 ;       APLOO6 < ...,APLOO6,APLOP
     11                                 ;
     12
     13          000006                OVROPR= 6                       ;OVERLAY SPECIFIED OPERATORS
```

```
      5
      6                                 ;++
      7                                 ; FUNCTIONAL DESCRIPTION:
      8                                 ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
      9                                 ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
     10                                 ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
     11                                 ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
     12                                 ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
     13                                 ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
     14                                 ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
     15                                 ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
     16                                 ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
     17                                 ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
     18                                 ;       INTACT!
     19                                 ;--
     24
     27
    124
    125                                 ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
    126                                 ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
    127                                 ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
    128                                 ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
    129                                 ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
    130                                 ;MACROS.
    131
    132                                 .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
    133                                 .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
    134                                 .IF EQ OVROPR-D.'NAME
    135                                         .PSECT  CODE
    136                                 D$'NAME::
    137                                 .ENDM
    138
    139                                 .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
    140                                 .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
    141                                 .IF EQ OVROPR-M.'NAME
    142                                         .PSECT  CODE
    143                                         .IF NE NUM
    144                                 M$'NAME::
    145                                     .ENDC
    146                                 .ENDM
    147
    148                                 .MACRO  LIST
    149                                         .ENDC
    150                                 .ENDM
    151
    152                                 .MACRO  PRMRES
    153                                         .IF EQ 1
    154                                 .ENDM
    155
    159                                                 ;************ ASSEMBLING THE FILE!!!!!
    160 007400                          PRMRES
    199 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 47
DEFINITIONS     ERRORS

    201 007400                          DOPINS  PLUS,2,T.SOP,.INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,.INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,.INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,.FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
APLOO6  MACRO M1113  08-FEB-80 09:33  PAGE 47-1
DEFINITIONS     ERRORS

    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

DEFINITIONS ERRORS

```
    641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
    652 007400                                  LIST
    653
    654
    655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
    662 007400                                  LIST
    663
    664
    665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
    676 007400                                  LIST
    677
    678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
    687 007400                                  LIST
    688
    689
    690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
    700 007400                                  LIST
    701
    702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    712 007400                                  LIST
    713
    714
    715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    733 007400                                  LIST
    734
    735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    745 007400                                  LIST
    746
    747
    748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
    758 007400                                  LIST
    759
    760
    761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
    780 007400                                  LIST
    781
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 49
DEFINITIONS     ERRORS

    783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    786 007400                                  LIST
    787
    788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    791 007400                                  LIST
    792
    793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
    799 007400                                  LIST
    800
    801
    802 007400                          MOPINS  $IDN,-2
    808 007400                                  LIST
    809
    810
    811 007400                          MOPINS  $FLR,-14
    855 007400                                  LIST
    856
    857
    858 007400                          MOPINS  $CEI,-16
    868 007400                                  LIST
    869
    870
    871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
    878 007400                                  LIST
    879
    880 007400                          MOPINS  $NEG,-4
    887 007400                                  LIST
    888
    889
    890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
    901 007400                                  LIST
    902
    903 007400                          MOPINS  $SIG,-6
    914 007400                                  LIST
    915
    916
    917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
    928 007400                                  LIST
    929
    930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
    939 007400                                  LIST
    940
    941
    942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
    949 007400                                  LIST
    950
    951
    952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
    959 007400                                  LIST
    960
    961 007400                          PRMRES
    966 007400                                  LIST
    967
    968
    969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
    977 007400                                  LIST
    978
```

```
    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 51
DEFINITIONS     ERRORS

   1189 007400                          MOPINS  SHAP,102
   1211 007400                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 54
DEFINITIONS     ERRORS

   1701 007400                          DOPINS  1COM,124
   1709 007400                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 007400                          DOPINS  COMP,60,T.DIM
   1789 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 57
DEFINITIONS     ERRORS

   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 58
DEFINITIONS     ERRORS

   2206 007400                          DOPINS  1ROT,146
   2214 007400                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 007400                          DOPINS  ROT,144,T.DIM
   2328 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 60
DEFINITIONS     ERRORS

   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 61
DEFINITIONS     ERRORS

   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 63
DEFINITIONS     ERRORS

   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 64
DEFINITIONS     ERRORS

   3221 007400                          MOPINS  MMD,122
   3316 007400                                  LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 65
DEFINITIONS     ERRORS

   3318 007400                          DOPINS  DMD,122
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 66
DEFINITIONS     ERRORS

   3864 007400                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 67
DEFINITIONS     ERRORS

   3866 007400                          DOPINS  MEM,70
   3921 007400                                  LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 68
DEFINITIONS     ERRORS

   3923 007400                                  MOPINS  NUB,106
   3924                                         .SBTTL  MIXED FUNCTION          NUB - MONADIC UP U - ELIMINATE DUPLICATES
   3925
   3926 000000  013704  003120'                        MOV     MONBLK,R4       ;POINT TO ARG
   3927 000004  016400  000004                        MOV     A.ALEN(R4),R0   ;GET LENGTH
   3928 000010  010002                                MOV     R0,R2           ;COPY IT
   3929 000012  013703  003124'                        MOV     MONTYP,R3       ;Get Arg Type
   3930 000016                                        ASH     SIZTAB(R3),R0   ;Compute Result Size
   3931 000032                                        PUSH    #A.DIM1+2       ;Save Offset/Header Size
   3932 000036  061600                                ADD     (SP),R0         ;ADD IN HEADER SIZE
   3933 000040                                        CALL    GETFRA          ;Get result, SI frame for it
   3934 000044  062620                                ADD     (SP)+,(R0)+     ;Add Offset
   3935 000046  112720  000001                        MOVB    #1,(R0)+        ;1 DIMENSION ONLY
   3936 000052  110310                                MOVB    R3,(R0)         ;LOAD BITS
   3937 000054  001003                                BNE     1$              ;Is it Character?
   3938 000056  042765  002000  177774                BIC     #T.NUM,SI.BITS(SI) ;Yes, Say So!
   3939 000064  152720  000200              1$:       BISB    #A.ASB,(R0)+        ;Indicate Array
   3940 000070  005020                                CLR     (R0)+           ;NO LENGTH YET
   3941 000072  005020                                CLR     (R0)+           ;NO DIM SIZE EITHER
   3942 000074  005702                                TST     R2              ;ANY LENGTH?
   3943 000076  001476                                BEQ     DONE            ;NO
   3944 000100  016337  000000G 005122' NUB1:         MOV     COMCT(R3),COMPTR    ;Compare Routine
   3945 000106  016337  000054' 003160'               MOV     LENTAB(R3),LARGFL   ;Left Arg Increment
   3946 000114  016337  000054' 003202'               MOV     LENTAB(R3),RARGFL   ;Right Arg, Too
   3947 000122                                        MUL     #5.,R3              ;Times 5 for Lookup
   3948 000136  016337  000000G 003176'               MOV     LARGCT(R3),LARGCV   ; Left Arg Fetch
   3949 000144  016337  000000G 003220'               MOV     RARGCT(R3),RARGCV   ; Right Arg Fetch
   3950 000152  016337  000000G 003152'               MOV     RESCT(R3),RESCVT;Result Store
   3951 000160  005046                                CLR     -(SP)           ;SAVE SP WORD
   3952 000162  010001                                MOV     R0,R1           ;Initialize Result Pointer
   3953 000164  010200                                MOV     R2,R0           ;Set Count
   3954 000166  013702  003122'                        MOV     MONPTR,R2       ;POINT TO DATA
   3955 000172  010137  003140'                        MOV     R1,RESPTR       ;SAVE POINTER TO START OF RESULT DATA
   3956                                         ;
   3957                                         ;CLEAR DUPLICATE OUT OF LITTERAL ARRAY
   3958                                         ;
   3959 000176  013703  003140'              CLROUT: MOV     RESPTR,R3       ;POINT TO START OF RESULT
   3960 000202                                        CALL    @LARGCV         ;Fetch Left Arg
   3961
   3962 000206  020301                        1$:     CMP     R3,R1           ;DONE COMPARING?
   3963 000210  001406                                BEQ     2$              ;YES
   3964 000212                                        CALL    @RARGCV         ;Fetch Previous Value
   3965 000216                                        CALL    @COMPTR         ;Is it the Same?
   3966 000222  001413                                BEQ     3$              ;YES
   3967 000224  000770                                BR      1$              ;TRY SOME MORE
   3968
   3969 000226  013737  003200' 003154' 2$:           MOV     LARGVA,RESVAL   ;Copy Arg
   3970 000234                                        STF     AC1,AC0         ;Whatever it is
   3971 000242                                        CALL    @RESCVT         ;Store Result
   3972 000246  104402                                SYSERR                  ;***Impossible***
   3973 000250  005216                                INC     (SP)            ;SAY ONE MORE CHR FOUND FOR RESULT
   3974 000252                                3$:     SOB     R0,CLROUT       ;TEST ALL CHRS
   3975 000256  000400                                BR      4$              ;FINISH UP
   3976                                         ;
   3977                                         ;CLEAN UP RESULT
   3978                                         ;
   3979 000260  016500  177776                4$:     MOV     SI.VAL(SI),R0   ;POINT TO RESULT
```

MIXED FUNCTION NUB - MONADIC UP U - ELIMINATE DUPLICATES

```
3980 000264  062700  000004                    ADD     #A.ALEN,R0      ;POINT TO LENGTH
3981 000270                                     POP     (R0)            ;LOAD PROPER LENGTH
3982 000272  012010                             MOV     (R0)+,(R0)      ;AND PROPER DIMSIZE
3983 000274                              DONE:  CALL    CARVE           ;CARVE UP RESULT VALUE
3984 000300                                     RETURN                  ;DONE
3986 000302                                     LIST
3987
3988
3989
3990
3991
3992 000302                              DOPINS UNION,106
3993                                     .SBTTL MIXED FUNCTION          UNION
3994 000302  012737  000001  005066'             MOV     #1,OPTMP0       ;Mark Us the Union
3995 000310                                     CALL    SETUP           ;Setup the fetch parameters and get some core
3996 000314  010001                             MOV     R0,R1           ;Copy pointer to result data
3997 000316  013700  003166'                    MOV     LARGLN,R0       ;Get the length of the Left argument
3998 000322  010004                             MOV     R0,R4           ;Copy the length
3999 000324  001416                             BEQ     3$              ;The Left argument is null
4000 000326  011302                             MOV     (R3),R2         ;Point to  the Left argument data
4001 000330                              1$:    CALL    @LARGCV         ;Copy the left argument to the result
4002 000334  013737  003200' 003154'            MOV     LARGVA,RESVAL   ;Move the left argument to the result
4003 000342                                     STF     AC1,AC0         ;Floating point too
4004 000350                                     CALL    @RESCVT         ;Store it into the result array
4005 000354  104402                             SYSERR                  ;Impossible
4006 000356                                     SOB     R0,1$           ;Loop for all the Left argument elements
4007
4008 000362  013700  003210'             3$:    MOV     RARGLN,R0       ;Get the length of the Right argument
4009 000366  001423                             BEQ     8$              ;The Right argument is null
4010 000370  060004                             ADD     R0,R4           ;Result length equals sum of argument lengths
4011 000372  013703  003206'                    MOV     RARGPT,R3       ;Point to Right argument data
4012 000376  013737  003150' 003160'            MOV     RESLEN,LARGFL   ;Get result element size
4013 000404                              4$:    CALL    @RARGCV         ;Get a Right argument element
4014 000410  013737  003222' 003154'            MOV     RARGVA,RESVAL   ;Copy its value to the result
4015 000416                                     STF     AC2,AC0         ;Floating point too
4016 000424                                     CALL    @RESCVT         ;Store it into the result array
4017 000430  104402                             SYSERR                  ;Impossible
4018 000432                                     SOB     R0,4$           ;Loop for all the Right argument elements
4019
4020 000436  000137  000712'             8$:    JMP     SETOP5          ;Finish up
4022 000442                                     LIST
4023
4024
4025
4026
4027
4028 000442                              DOPINS INTR,66
4029                                     .SBTTL MIXED FUNCTION          INTERSECTION
4030
4031 000442  005037  005066'                    CLR     OPTMP0          ;Mark Us as Intersection
4032 000446                                     CALL    SETUP           ;Setup argument fetch parameters and result
4033 000452  012737  000660' 005070'            MOV     #SETOP3,OPTMP1  ;Setup branch targets for comparison routine
4034 000460  012737  000704' 005072'            MOV     #SETOP4,OPTMP2
4035 000466                                     CALL    SETOP           ;Go do the intersection
4036 000472  013700  003140'                    MOV     RESPTR,R0       ;Get a pointer to the result data
4037 000476  010037  003122'                    MOV     R0,MONPTR       ;Copy the result pointer for NUB
4038 000502  013703  003144'                    MOV     RESTYP,R3       ;Get the result type
```

```
4039 000506 013702 005076'                  MOV     OPTMP4,R2       ;The size of the result
4040 000512 000137 000100'                  JMP     NUB1            ;Now use NUB to delete duplicates
4042 000516                                 LIST
4043
4044
4045
4046
4047
4048 000516                         DOPINS  SDIF,50
4049                                .SBTTL  MIXED FUNCTION          SET DIFFERENCE
4050
4051 000516 012737 177777 005066'           MOV     #-1,OPTMP0      ;Mark Us as Different
4052 000524                                 CALL    SETUP           ;Setup argument fetch parameters and result
4053 000530 012737 000704' 005070'          MOV     #SETOP4,OPTMP1  ;Setup branch targets for comparison routine
4054 000536 012737 000660' 005072'          MOV     #SETOP3,OPTMP2
4055
4056
4057 000544 010001                  SETOP:  MOV     R0,R1           ;Copy the result pointer
4058 000546 011302                          MOV     (R3),R2         ;Point to the Left argument data
4059 000550 005004                          CLR     R4              ;The result has no length yet
4060 000552 005737 003166'                  TST     LARGLN          ;Is the Left argument null?
4061 000556 001455                          BEQ     SETOP5          ;Yes, Then the Result is null
4062 000560 005737 005066'                  TST     OPTMP0          ;Are we doing Intersection?
4063 000564 001003                          BNE     1$              ;No
4064 000566 005737 003210'                  TST     RARGLN          ;Yes, Is the Right argument null?
4065 000572 001447                          BEQ     SETOP5          ;Yes, Then the Result is null
4066 000574 013703 003206'          1$:     MOV     RARGPT,R3       ;Point to the Right argument data
4067 000600 013737 003216' 003202'          MOV     RARGLE,RARGFL   ;Get the Right argument element size
4068 000606                         SETOP1: CALL    @LARGCV         ;Get the next Left argument
4069 000612 013700 003210'                  MOV     RARGLN,R0       ;Get length of Right argument
4070 000616 001420                          BEQ     SETOP3          ;Right Argument is null
4071 000620 013703 003206'                  MOV     RARGPT,R3       ;Point to the Right argument data
4072 000624 013737 003216' 003202'          MOV     RARGLE,RARGFL   ;Get the Right argument element size
4073 000632                         SETOP2: CALL    @RARGCV         ;Get a Right argument element
4074 000636                                 CALL    @COMPTR         ;Are the Left and Right arguments the same?
4075 000642 001002                          BNE     1$              ;No
4076 000644 000177 005070'                  JMP     @OPTMP1         ;Yes, Insect: Copy left arg, Diff: Next left arg
4077 000650                         1$:     SOB     R0,SETOP2       ;Then loop for the next Right argument
4078 000654 000177 005072'                  JMP     @OPTMP2         ;Insect: Next left arg, Diff: Copy left arg
4079
4080 000660 013737 003200' 003154' SETOP3: MOV     LARGVA,RESVAL   ;Copy the Left argument to the result
4081 000666                                 STF     AC1,AC0         ;Floating point type too
4082 000674                                 CALL    @RESCVT         ;Store it into the result
4083 000700 104402                          SYSERR                  ;This is impossible
4084 000702 005204                          INC     R4              ;One more element in the result
4085
4086 000704 005337 003166'          SETOP4: DEC     LARGLN          ;Loop on the Left argument
4087 000710 003336                          BGT     SETOP1          ;Until done
4088
4089 000712 013700 003136'          SETOP5: MOV     RESULT,R0       ;Get pointer to result block
4090 000716 010460 000004                   MOV     R4,A.ALEN(R0)   ;Fill in the length of the result
4091 000722 010460 000006                   MOV     R4,A.DIM1(R0)   ;And the dimension
4092 000726 010437 005076'                  MOV     R4,OPTMP4       ;Save the result size
4093 000732                                 CALL    CARVE           ;Carve out the extra space from the result blk
4094 000736                                 RETURN                  ;All done
4095                                ;+
4096                                ; This routine sets up the argument fetching and conversion routines for
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 68-3
MIXED FUNCTION          SET DIFFERENCE

                                        ; Union and Intersection.  It also creates an array block to hold the result
                                        ;-

000740                          SETUP:  CALL    GETREG
000744  042700  177771                 BIC     #^C<A.NUM>,R0   ;Clear out garbage bits for left argument
000750  042701  177771                 BIC     #^C<A.NUM>,R1   ;Clear out garbage bits for right argument
000754  010002                         MOV     R0,R2           ;Assume the left argument type is bigger
000756  010104                         MOV     R1,R4           ;Assume the right argument type is smaller
000760  020001                         CMP     R0,R1           ;Are the arguments the same type?
000762  001423                         BEQ     4$              ;Yes
000764  010003                         MOV     R0,R3           ;No, Copy the Left argument type
000766                                 MUL     R1,R3           ;Are both arguments numeric?
001000  001001                         BNE     1$              ;Yes
001002  104417                         DOMERR                  ;No, Characters and numerics don't mix
001004  020100                 1$:     CMP     R1,R0           ;Which type really is bigger?
001006  002402                         BLT     2$              ;The left is bigger
001010  010102                         MOV     R1,R2           ;The right is bigger, Get the compute type
001012  010004                         MOV     R0,R4           ;Get the result type too
001014  005737  005066'         2$:     TST     OPTMP0          ;What are we doing?
001020  001404                         BEQ     4$              ;Intersection, Use the smaller type
001022  003002                         BGT     3$              ;Union
001024  010004                         MOV     R0,R4           ;Difference, Use the Left type
001026  000401                         BR      4$
001030  010204                 3$:     MOV     R2,R4           ;Union, Use the bigger type
001032  010437  003144'         4$:     MOV     R4,RESTYP       ;Store the result type
001036  016437  000044' 003146'         MOV     SIZTAB(R4),RESSFT  ;Get the Shift field for the result
001044  016437  000054' 003150'         MOV     LENTAB(R4),RESLEN  ;Get the length of a result element
001052  016237  000000G 005122'         MOV     COMCT(R2),COMPTR   ;Pointer to Comparison routine
001060  010203                         MOV     R2,R3           ;Copy the compute type
001062  006303                         ASL     R3              ;Get the pointer to the result store rtn
001064  006303                         ASL     R3              ;Compute type times four
001066  060300                         ADD     R3,R0           ; plus Left type
001070  016037  000000G 003176'         MOV     LARGCT(R0),LARGCV  ;Point to left argument fetch routine
001076  060301                         ADD     R3,R1           ; plus Right type
001100  016137  000000G 003220'         MOV     RARGCT(R1),RARGCV  ;Point to right argument fetch routine
001106  006302                         ASL     R2              ;Compute type times four
001110  006302                         ASL     R2
001112  060402                         ADD     R4,R2           ; plus Result type
001114  016237  000000G 003152'         MOV     RESCT(R2),RESCVT   ;Point to result store routine
001122  013703  003162'                 MOV     LARGBL,R3       ;Point to left argument
001126  013704  003204'                 MOV     RARGBL,R4       ;Point to right argument
001132  016300  000004                 MOV     A.ALEN(R3),R0   ;Get length of left argument
001136  005737  005066'                 TST     OPTMP0          ;Doing Union, Intersection, or Difference?
001142  003402                         BLE     6$              ;Diff. or Inter., Use the Left argument size
001144  066400  000004                 ADD     A.ALEN(R4),R0   ;Union, Add in length of the right argument
001150  010002                 6$:     MOV     R0,R2           ;Copy the result length
001152                                 ASH     RESSFT,R0       ;Multiply by the result element size
001166  062700  000010                 ADD     #A.DIM1+2,R0    ;Add in header overhead
001172                                 CALL    GETFRA          ;Get result, SI frame for it
001176  010037  003136'                 MOV     R0,RESULT       ;Save pointer to result block
001202  062720  000010                 ADD     #A.DIM1+2,(R0)+ ; Pointer to value
001206  112720  000001                 MOVB    #1,(R0)+        ; Result is always a vector
001212  113710  003144'                 MOVB    RESTYP,(R0)     ; The Bits
001216  001003                         BNE     7$              ; It's Numeric so SI frame is Ok
001220  042765  002000  177774         BIC     #T.NUM,SI.BITS(SI)  ;Fix SI frame bits
001226  152720  000200         7$:     BISB    #A.ASB,(R0)+    ; Make it an array block
001232  005020                         CLR     (R0)+           ; Don't know true length yet
```

```
4154 001234  005020                          CLR     (R0)+           ; Nor dimension
4155 001236  005702                          TST     R2              ;Both arguments null?
4156 001240  001002                          BNE     8$              ;No, Just return
4157 001242  062706  000002                  ADD     #2,SP           ;Yes, junk return address and return
4158 001246  010037  003140'          8$:     MOV     R0,RESPTR       ;Save pointer to result value
4159 001252                                  RETURN
4161 001254                                  LIST
```

```
                                        DOPINS  SCON,234
                                        .SBTTL  MIXED FUNCTION          LEFT U - STRICT CONTAINMENT
                                        ;
                                        ; OPTMP0: 0 - Possibly equal set
                                        ;         1 - Proper set
                                        ;
                                        ; OPTMP1: 0 - First time through subset test code
                                        ;         1 - Second time through subset test code
                                        ;
                                        ; OPTMP2: 0 - Subset function
                                        ;         1 - Containment function
001254  012737  000001  005066'                 MOV     #1,OPTMP0       ;Setup for proper containment
001262  000402                                  BR      CONTNS          ;Now go do it
001264                                          LIST


                                        DOPINS  CONT,112
                                        .SBTTL  MIXED FUNCTION          UNDERSCORE LEFT U - CONTAINMENT
                                        ;
                                        ; RETURNS A 1 IF EVERY ELEMENT ON RIGHT IS PRESENT IN LEFT ARG.
                                        ; RETURNS 0 OTHERWISE.
                                        ;
001264  005037  005066'                         CLR     OPTMP0          ;Setup for ordinary containment
001270  005037  005070'                 CONTNS: CLR     OPTMP1          ;Clear cycle flag
001274  012737  000001  005072'                 MOV     #1,OPTMP2       ;Setup for containment function
001302                                  CONTN1: CALL    GETREG          ;Get pointers to arguments
001306  010402                                  MOV     R4,R2           ;Swap pointers to arguments around
001310  010304                                  MOV     R3,R4           ;So a common subset routine can be used
001312  010203                                  MOV     R2,R3
001314  010102                                  MOV     R1,R2           ;Swap the bits too
001316  010001                                  MOV     R0,R1
001320  010200                                  MOV     R2,R0
001322  000414                                  BR      SUBST2          ;Now go test for a subset
001324                                          LIST


                                        DOPINS  PSUB,236
                                        .SBTTL  MIXED FUNCTION          RIGHT U - PROPER SUBSET OF

001324  012737  000001  005066'                 MOV     #1,OPTMP0       ;Setup for proper subset
001332  000402                                  BR      SUBSET          ;Now go do it
001334                                          LIST


                                        DOPINS  SUBS,116
                                        .SBTTL  MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

                                        ;
                                        ; RETURNS 1 IF RIGHT ARG CONTAINS LEFT ARG, ELSE RETURNS 0
                                        ;

001334  005037  005066'                         CLR     OPTMP0          ;Setup for ordinary subset function
001340  005037  005070'                 SUBSET: CLR     OPTMP1          ;Clear cycle flag
001344  005037  005072'                         CLR     OPTMP2          ;Setup for subset function
001350                                  SUBST1: CALL    GETREG          ;Place argument pointers and bits into registers
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 69-1
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

 4223 001354  042700  177771          SUBST2: BIC     #^C<A.NUM>,R0   ;Clear out Garbage Bits
 4224 001360  042701  177771                  BIC     #^C<A.NUM>,R1   ;Me Too! Me Too!
 4225 001364  010002                          MOV     R0,R2           ;Get An Arg
 4226 001366  020102                          CMP     R1,R2           ;Who is Larger?
 4227 001370  003401                          BLE     1$              ;The one we guessed, of course
 4228 001372  010102                          MOV     R1,R2           ;Get the Arg
 4229 001374  016237  000000G 005122' 1$:     MOV     COMCT(R2),COMPTR ;Compare Routine
 4230 001402  006302                          ASL     R2              ;Times 2
 4231 001404  006302                          ASL     R2              ;Times 4
 4232 001406  016037  000054' 003160'         MOV     LENTAB(R0),LARGFL ;Left Increment
 4233 001414  016137  000054' 003202'         MOV     LENTAB(R1),RARGFL ;Right Increment
 4234 001422  060201                          ADD     R2,R1           ;2-Dimesional
 4235 001424  060200                          ADD     R2,R0           ;  Lookups
 4236 001426  016037  000000G 003176'         MOV     LARGCT(R0),LARGCV ;Left Fetch
 4237 001434  016137  000000G 003220'         MOV     RARGCT(R1),RARGCV ;Right Fetch
 4238 001442                                  PUSH    A.ALEN(R3)      ;Push the length of the Left argument
 4239 001446  001450                          BEQ     6$              ;The Left argument is the Null set
 4240 001450                                  PUSH    A.ALEN(R4)      ;Push the length of the Right argument
 4241 001454  001457                          BEQ     7$              ;The right argument is the Null set
 4242 001456  011302                          MOV     (R3),R2         ;Point to the Left argument
 4243 001460  011404                          MOV     @R4,R4          ;Point to the right argument
 4244 001462                          2$:     CALL    @LARGCT         ;Get Left argument
 4245 001466  011600                          MOV     (SP),R0         ;Reset the Right argument count
 4246 001470  010403                          MOV     R4,R3           ;Reset the right argument pointer
 4247
 4248 001472                          3$:     CALL    @RARGCV         ;Get Right argument
 4249 001476                                  CALL    @COMPTR         ;Do Compare
 4250 001502  001411                          BEQ     5$              ;Found it
 4251 001504                                  SOB     R0,3$           ;Not found yet
 4252                                                                  ;End of Right argument
 4253 001510  005737  005070'                 TST     OPTMP1          ;Been through here before?
 4254 001514  001437                          BEQ     7$              ;No
 4255 001516  012737  000001  003154' 4$:     MOV     #1,RESVAL       ;Yes, return TRUE value
 4256 001524  000435                          BR      8$              ;Go finish up
 4257
 4258 001526  005366  000002          5$:     DEC     2(SP)           ;End of the Left argument?
 4259 001532  003353                          BGT     2$              ;No
 4260 001534  005737  005066'                 TST     OPTMP0          ;Yes, Looking for a proper set?
 4261 001540  001766                          BEQ     4$              ;No
 4262 001542  005737  005070'                 TST     OPTMP1          ;Yes, Been through subset test code?
 4263 001546  001022                          BNE     7$              ;Yes
 4264 001550  012737  000001  005070'         MOV     #1,OPTMP1       ;No, Then set cycle flag
 4265 001556  022626                          CMP     (SP)+,(SP)+     ;Clear counts off the stack for second run thru
 4266 001560  005737  005072'                 TST     OPTMP2          ;Are we doing a Subset or Containment function?
 4267 001564  001646                          BEQ     CONTN1          ;Containment
 4268 001566  000670                          BR      SUBST1          ;Subset, Now try it the other way around
 4269                                                                  ;Left argument is Null
 4270 001570  005046          6$:     CLR     -(SP)           ;So we have something to pop later
 4271 001572  012737  000001  003154'         MOV     #1,RESVAL       ;Set Result to "TRUE"
 4272 001600  005737  005066'                 TST     OPTMP0          ;Looking for a proper set?
 4273 001604  001405                          BEQ     8$              ;No
 4274 001606  005764  000004                  TST     A.ALEN(R4)      ;Yes, Is the right set Null?
 4275 001612  001002                          BNE     8$              ;No
 4276 001614  005037  003154'          7$:     CLR     RESVAL          ;Yes, Return "FALSE" value
 4277
 4278 001620  012700  000012          8$:     MOV     #A.DIM1+4,R0    ;Get size of Result block
 4279 001624                                  CALL    GETFRA          ;Get result block and SI frame for it
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 69-2
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

   4280 001630  062720  000010                ADD     #A.DIM1+2,(R0)+ ;Offset to data
   4281 001634  012720  101001                MOV     #<A.ASB+.BOOLEAN>*400+1,(R0)+   ;Result is Boolean vector
   4282 001640  012720  000001                MOV     #1,(R0)+        ;A.ALEN
   4283 001644  012720  000001                MOV     #1,(R0)+        ;A.DIM1
   4284 001650  113720  003154'               MOVB    RESVAL,(R0)+    ;Put in result value
   4285 001654  022626                        CMP     (SP)+,(SP)+     ;Clean counts off stack
   4286 001656                                RETURN                  ;All done
   4288 001660                                LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 70
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

   4290 001660                              MOPINS  EXE,70
   4336 001660                                      LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 71
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

   4338 001660                              MOPINS  ENCD,76
   4411 001660                                      LIST
```

```
 4413 001660                            MOPINS  FMT,160
 4423 001660                                    LIST
 4424
 4425
 4426
 4427
 4428
 4429 001660                            DOPINS  FMT,160
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 73
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

   4953 001660                                      LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 74
MIXED FUNCTION          UNDERSCORE RIGHT U - SUBSET OF

   4955 001660                          MOPINS  IBM,132,,,MONIBM
   4958 001660                                  LIST
   4959
   4960
   4961 001660                          DOPINS  IBM,132
   5046 001660                                  LIST
```

```
 5048 001660                            MOPINS  QUAD,74,T.NIL+T.SPOP
 5075 001660                                    LIST
 5076
 5077
 5078
 5079
 5080
 5081 001660                            DOPINS  QUAD,74,T.NIL+T.SPOP
 5097 001660                                    LIST
 5098
 5099
 5100
 5101
 5102
 5103 001660                            MOPINS  QQAD,142,T.NIL+T.SPOP
 5135 001660                                    LIST
 5136
 5137
 5138
 5139
 5140
 5141 001660                            DOPINS  QQAD,142,T.NIL+T.SPOP
 5152 001660                                    LIST
 5153
 5154
 5155
 5156
 5157
 5158 001660                            MOPINS  DQAD,134,T.NIL+T.SPOP
 5159                                   .SBTTL  MIXED FUNCTION          DEL QUAD INPUT
 5160
 5161 001660                                    PUSH    INPTR           ;SAVE CURRENT INPUT POINER
 5162 001664  012737  000000G 000064'          MOV     #GETL,INPTR     ;INPUT ALWAYS COMES FROM KEYBOARD
 5163 001672                                    CALL    GETLIN          ;GET LINE OF INPUT
 5164 001676                                    POP     INPTR           ;RESTORE INPUT POINTER
 5165 001702  010100                            MOV     R1,R0           ;Copy Pointer past the end
 5166 001704  162700  003316'                   SUB     #LBUFF,R0       ;   minus Start of buffer
 5167 001710  122741  000015                    CMPB    #CHR.CR,-(R1)   ;Previous character a <CR>?
 5168 001714  001001                            BNE     1$              ;Nope
 5169 001716  005300                            DEC     R0              ;Yes, go to it
 5170 001720  010002                    1$:     MOV     R0,R2           ;COPY LEN
 5171 001722  062700  000010                    ADD     #A.DIM1+2,R0    ;GET BLOCK SIZE
 5172 001726                                    CALL    GETFRA          ;AND GET THE BLOCK
 5173 001732  042765  002000  177774           BIC     #T.NUM,SI.BITS(SI)
 5174                                                                   ;Clear the Numeric Bit
 5175 001740  062720  000010                    ADD     #A.DIM1+2,(R0)+ ;Add OFFSET
 5176 001744  012720  100001                    MOV     #<A.ASB+.CHARACTER>*400+1,(R0)
 5177                                                                   ;1 DIM, NO BIT
 5178 001750  010220                            MOV     R2,(R0)+        ;LEN=SIZE OF ONE
 5179 001752  010220                            MOV     R2,(R0)+        ;SAME WI   FIRST DIM
 5180 001754  001415                            BEQ     4$              ;CHECK FOR EMPTY ARRAY
 5181 001756  012701  003316'                   MOV     #LBUFF,R1       ;POINT TO START OF LINE BUFFER
 5182
 5183 001762  112103                    2$:     MOVB    (R1)+,R3        ;GET CHAR
 5184 001764  105737  000403'                   TSTB    TMODE           ;APL MODE?
 5185 001770  001404                            BEQ     3$              ;YES, SO NO TRANSLATION
 5186 001772  042703  177400                    BIC     #<UPPER>,R3     ;Clear upper bits
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 75-1
MIXED FUNCTION          DEL QUAD INPUT

   5187 001776  116303  000000G                    MOVB    TRNTAB(R3),R3   ;GET TRANSLATED CHARACTER
   5188 002002  110320                     3$:     MOVB    R3,(R0)+        ;STORE CHARACTER
   5189 002004                                     SOB     R2,2$           ;ALL OF 'EM
   5190 002010                             4$:     RETURN
   5192 002012                                     LIST
   5193
   5194
   5195
   5196 002012                             DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
   5199 002012                                     LIST
```

```
 5201 002012                    MOPINS  BXAV,162,T.NIL
 5219 002012                            LIST
 5220
 5221
 5222
 5223
 5224
 5225 002012                    MOPINS  BXCT,164,T.NIL+T.SPOP
 5234 002012                            LIST
 5235
 5236
 5237 002012                    DOPINS  BXCT,164,T.NIL+T.SPOP
 5272 002012                            LIST
 5273
 5274
 5275
 5276
 5277
 5278 002012                    MOPINS  BXIO,166,T.NIL+T.SPOP
 5287 002012                            LIST
 5288
 5289
 5290 002012                    DOPINS  BXIO,166,T.NIL+T.SPOP
 5302 002012                            LIST
 5303
 5304
 5305
 5306
 5307 002012                    MOPINS  BXLC,212,T.NIL
 5340 002012                            LIST
 5341
 5342
 5343
 5344
 5345
 5346
 5347 002012                    MOPINS  BXPP,170,T.NIL+T.SPOP
 5356 002012                            LIST
 5357
 5358
 5359 002012                    DOPINS  BXPP,170,T.NIL+T.SPOP
 5381 002012                            LIST
 5382
 5383
 5384
 5385
 5386
 5387
 5388 002012                    MOPINS  BXPW,172,T.NIL+T.SPOP
 5397 002012                            LIST
 5398
 5399
 5400 002012                    DOPINS  BXPW,172,T.NIL+T.SPOP
 5417 002012                            LIST
 5418
 5419
 5420
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 76-1
MIXED FUNCTION          DEL QUAD INPUT

   5421
   5422
   5423 002012                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 002012                                  LIST
   5433
   5434
   5435 002012                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 002012                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 002012                          MOPINS  BXTS,200,T.NIL
   5562 002012                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 002012                          MOPINS  BXWA,176,T.NIL
   5592 002012                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 77
MIXED FUNCTION          DEL QUAD INPUT

   5594 002012                          MOPINS  BXCR,206
   5716 002012                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 78
MIXED FUNCTION          DEL QUAD INPUT

   5718 002012                          MOPINS  BXEX,202
   5775 002012                                  LIST
```

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 79
MIXED FUNCTION          DEL QUAD INPUT

   5777 002012                          MOPINS  BXFX,210
   5879 002012                                  LIST
```

B 1
C 1
D 1
E 1
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1
B 2
C 2
D 2
E 2
F 2
G 2
H 2
I 2
J 2
K 2
L 2
M 2
N 2
B 3
C 3
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5
I 5

APL004

J 5
K 5
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7
N 7
B 8
C 8
D 8
E 8
F 8
G 8
H 8
I 8
J 8
K 8
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9
J 9
K 9
L 9
M 9
N 9
B 10
C 10
D 10

APL005

E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14

APL006

M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 80
MIXED FUNCTION          DEL QUAD INPUT

   5881 002012                          MOPINS  BXNC,204
   5950 002012                                  LIST
```

```
APLOOo  MACRO M1113  08-FEB-80 09:33  PAGE 81
MIXED FUNCTION          DEL QUAD INPUT

   5952 002012                          MOPINS  BXNL,214
   5963 002012                                  LIST
   5964
   5965
   5966 002012                          DOPINS  BXNL,214
   6104 002012                                  LIST
```

```
 6106 002012                          DOPINS  FGET,150,T.DIM
 6330 002012                                  LIST
 6331
 6332
 6333
 6334
 6335
 6336 002012                          DOPINS  FPUT,152,T.DIM
 6485 002012                                  LIST
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 83
MIXED FUNCTION          DEL QUAD INPUT

  6487 002012                          DOPINS  FPNT,154,T.DIM
  6714 002012                                  LIST
```



```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 84
```

MIXED FUNCTION DEL QUAD INPUT

```
 6716 002012                            MOPINS  1SCN,130
 6723 002012                                    LIST
 6724
 6725
 6726 002012                            MOPINS  SCAN,62,T.DIM
 6732 002012                                    LIST
 6733
 6734
 6735 002012                            MOPINS  1RDT,124
 6741 002012                                    LIST
 6742
 6743
 6744 002012                            MOPINS  RDCT,60,T.DIM
 7044 002012                                    LIST
```

```
7046 002012                            MOPINS  GETIDN,0
7047                                   .SBTTL  GETIDN                  CREATE IDENTITY ELEMENT ARRAY
7048
7049                                   ;++
7050                                   ; FUNCTIONAL DESCRIPTION
7051                                   ;       This routine creates an array of identity elements for a given
7052                                   ;       scalar function.  It is called by Reduction/Scan and Inner
7053                                   ;       product.
7054                                   ;
7055                                   ; INPUTS:
7056                                   ;       R4      - Scalar function number
7057                                   ;       OPTMP5  - The reduction dimension
7058                                   ;       R1      - Number of elements in result
7059                                   ;       R2      - Pointer to Left or only argument
7060                                   ;       R3      - Pointer to Right argument or Zero
7061                                   ;--
7062 002012                                    .PSECT  CODE
7063
7064 002012                            GETIDN::
7065 002012                                    PUSH    <NUMLEN,R1,R2>  ;Save identity element size and some registers
7066 002022  012700  005104'                   MOV     #OPTMP7,R0      ;Point to temp area for floating identity
7067 002026  010002                            MOV     R0,R2           ;Copy the pointer
7068 002030  013701  000062'                   MOV     NUMLEN,R1       ;Get length of floating point numbers
7069 002034  112720  177777            1$:     MOVB    #-1,(R0)+       ;Create biggest negative floating point number
7070 002040                                    SOB     R1,1$           ;Loop for the whole length
7071 002044  012737  000006  003144'           MOV     #.FLOAT,RESTYP  ;Assume result type is floating point
7072 002052  013737  000052' 003146'           MOV     PRECSN,RESSFT   ;Get result shift count
7073 002060  005704                            TST     R4              ;Is the function number negative i.e. floating
7074 002062  100001                            BPL     2$              ;No
7075 002064  005404                            NEG     R4              ;Yes, Then make it positive
7076 002066  022704  000000G           2$:     CMP     #OP.CEI,R4      ;Is the function Ceiling?
7077 002072  001431                            BEQ     4$              ;Yes, Then use largest negative number
7078 002074  042712  100000                    BIC     #100000,(R2)    ;Convert identity to largest positive number
7079 002100  022704  000000G                   CMP     #OP.FLR,R4      ;Is the function Floor?
7080 002104  001424                            BEQ     4$              ;Yes, Then use largest floating number
7081 002106  006204                            ASR     R4              ;Convert function number to word pointer
7082 002110  116401  000000G                   MOVB    DYDBTL(R4),R1   ;Get function's bits
7083 002114  032701  000040                    BIT     #CS.IDN,R1      ;Does the function have an identity?
7084 002120  001001                            BNE     3$              ;Yes, Ok
7085 002122  104417                            DOMERR                  ;No, DOMAIN Error
7086 002124  012737  000002  003144' 3$:      MOV     #.BOOLEAN,RESTYP;Result type is Boolean (i.e. 1 or 0)
7087 002132  005037  003146'                   CLR     RESSFT          ;Setup result shift count
7088 002136  012766  000001  000004           MOV     #1,4(SP)        ;Setup size of one element
7089 002144  005012                            CLR     (R2)            ;Assume identity is zero
7090 002146  032701  000100                    BIT     #CS.ONE,R1      ;What is the identity really?
7091 002152  001401                            BEQ     4$              ;Zero, Good guess
7092 002154  005212                            INC     (R2)            ;One
7093
7094 002156                            4$:     POP     R2              ;Restore pointer to Left argument
7095 002160  116204  000002                    MOVB    A.DIMS(R2),R4   ;Get the left argument's rank
7096 002164                                    CONINC  R4              ;In case of a scalar
7097 002170                                    PUSH    R4              ;Save it for result rank
7098 002172  005316                            DEC     (SP)            ;Minus one for deleted dimension
7099 002174  005703                            TST     R3              ;Is there a right argument?
7100 002176  001410                            BEQ     5$              ;No
7101 002200  116301  000002                    MOVB    A.DIMS(R3),R1   ;Yes, Get its rank
7102 002204                                    CONINC  R1              ;In case of a scalar
```

GETIDN CREATE IDENTITY ELEMENT ARRAY

```
7103 002210  005301                          DEC     R1              ;Minus one for deleted dimension
7104 002212  060116                          ADD     R1,(SP)         ;Total result rank
7105 002214  010137  005066'                 MOV     R1,OPTMP0       ;Save right argument rank
7106 002220                          5$:     POP     R1              ;Save the result rank
7107 002222                                  PUSH    R3              ;Save the Right argument pointer
7108 002224                                  PUSH    R1              ;Restore the result rank
7109 002226  016603  000004                  MOV     4(SP),R3        ;Get size of result
7110 002232                                  CALL    GETSPA          ;Get space for result and fill in header
7111 002236                                  POP     R3              ;Are we reducing?
7112 002240  001410                          BEQ     11$             ;Yes
7113 002242  013701  005066'                 MOV     OPTMP0,R1       ;Are there any right dimensions?
7114 002246  001405                          BEQ     11$             ;No, Skip copying then
7115 002250  062703  000010                  ADD     #A.DIM1+2,R3    ;Point to second right dimension
7116
7117 002254  012320                  10$:    MOV     (R3)+,(R0)+     ;Copy the right argument dimension list
7118 002256                                  SOB     R1,10$          ;Fast
7119
7120 002262                          11$:    POP     <R4,R3>         ;Get result and element sizes
7121
7122 002266  010301                  12$:    MOV     R3,R1           ;Copy element size
7123 002270  012702  005104'                 MOV     #OPTMP7,R2      ;Get pointer to identity element
7124 002274  112220                  13$:    MOVB    (R2)+,(R0)+     ;Copy identity element
7125 002276                                  SOB     R1,13$          ;Real fast
7126 002302                                  SOB     R4,12$          ;Copy all the elements
7127
7128 002306                                  RETURN
7129
7130
7131                                 ;++
7132                                 ; FUNCTIONAL DESCRIPTION:
7133                                 ;       This routine gets space for a result array block, creates an SI
7134                                 ;       frame for it, and fills in the header part including the dimension
7135                                 ;       list.  It fills in all the dimensions of a simular array block
7136                                 ;       except a specified one.  It is currently called by GETIDN, Reduction,
7137                                 ;       and Inner product.
7138                                 ;
7139                                 ; INPUTS:
7140                                 ;       Top of SP - Rank of result (Popped off before return)
7141                                 ;       R2        - Pointer to source argument (Left)
7142                                 ;       R3        - Size of result
7143                                 ;       R4        - Rank of source argument (Left)
7144                                 ;       OPTMP5    - Pointer to source argument dimension to skip
7145                                 ;
7146                                 ; OUTPUTS:
7147                                 ;       R0        - Pointer to next word of result block
7148                                 ;       R3        - Size of result
7149                                 ;       All other registers modified
7150                                 ;--
7151
7152 002310                          GETSPA::
7153 002310                                  POP     R1              ;Swap return address
7154 002312                                  PUSH    (SP)            ; and result rank
7155 002314  010166  000002                  MOV     R1,2(SP)        ;  around
7156 002320  010301                          MOV     R3,R1           ;Get number of elements in result
7157 002322                                  ASH     RESSFT,R1       ;in bytes
7158 002336  102452                          BVS     5$              ;Too big, WS full
7159 002340  011600                          MOV     (SP),R0         ;Get result's rank
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 85-2
GETIDN                CREATE IDENTITY ELEMENT ARRAY

 7160 002342                                    CONINC  R0              ;In case of a scalar
 7161 002346  006300                            ASL     R0              ;Number of bytes in dimension list
 7162 002350  062700  000006                    ADD     #A.DIM1,R0      ;Overhead of array block
 7163 002354                                    PUSH    R0              ;Save offset
 7164 002356  060100                            ADD     R1,R0           ;Space for data
 7165 002360                                    CALL    GETFRA          ;Go get the space
 7166 002364  010037  003136'                   MOV     R0,RESULT       ;Save pointer to result block
 7167 002370  062610                            ADD     (SP)+,(R0)      ;Fill in result header, Pointer to data
 7168 002372  012037  003140'                   MOV     (R0)+,RESPTR    ;Save pointer
 7169 002376  111620                            MOVB    (SP),(R0)+      ;Rank
 7170 002400  113710  003144'                   MOVB    RESTYP,(R0)     ;Type
 7171 002404  001003                            BNE     1$              ;Is it numeric?
 7172 002406  042765  002000  177774            BIC     #T.NUM,SI.BITS(SI) ;No, Fixup SI frame bits
 7173 002414  152720  000200          1$:       BISB    #A.ASB,(R0)+    ;Bits, Array block
 7174 002420  010320                            MOV     R3,(R0)+        ;ALEN
 7175 002422  010320                            MOV     R3,(R0)+        ;Cheat for scalars and vectors
 7176 002424  022627  000001                    CMP     (SP)+,#1        ;What is the result rank?
 7177 002430  003414                            BLE     4$              ;It's a Scalar or vector
 7178 002432  005704                            TST     R4              ;Array, Any Left argument dimensions?
 7179 002434  001412                            BEQ     4$              ;No
 7180 002436  005740                            TST     -(R0)           ;Back up result dimension list pointer
 7181 002440  062702  000006                    ADD     #A.DIM1,R2      ;Point to Left argument dimension list
 7182
 7183 002444  020237  005100'          2$:       CMP     R2,OPTMP5       ;Is this the dimension?
 7184 002450  001401                            BEQ     3$              ;Yes, Skip it
 7185 002452  011220                            MOV     (R2),(R0)+      ;Copy this dimension
 7186 002454  005722                    3$:     TST     (R2)+           ;Next dimension
 7187 002456                                    SOB     R4,2$           ;Loop for all the diemnsions
 7188
 7189 002462                            4$:     RETURN
 7190
 7191 002464  104403                    5$:     WSFULL
 7193 002466                                    LIST
```



```
APLO06  MACRO M11    08-FEB-80 09:33  PAGE 86
```

```
APLO06  MACRO M11    08-FEB-80 09:33  PAGE 86
GETIDN                  CREATE IDENTITY ELEMENT ARRAY

   7195 002466                          MOPINS  SATAN.O
   7404 002466                                  LIST
```



```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 87
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 87
GETIDN                  CREATE IDENTITY ELEMENT ARRAY

   7406 002466                          MOPINS  OUTR,156
   7476 002466                                  LIST
```



```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 88
GETIDN                  CREATE IDENTITY ELEMENT ARRAY
```

```
7478 002466                            MOPINS  DOUTER,0
7479                                   .SBTTL  DOUTER                  PERFORMS THE ACTUAL OUTER PRODUCT OPERATION
7480
7481                                   ;+
7482                                   ;  Routine to do Outer product
7483                                   ;-
7484 002466                            DOUTER::
7485 002466  013703  003206'                    MOV     RARGPT,R3       ;POINT TO RIGHT DATA
7486 002472  013700  003210'                    MOV     RARGLN,R0       ;GET LEN OF RIGHT
7487
7488 002476                            1$:     CALL    @LARGCV         ;GET LEFT VALUE
7489 002502                            2$:     CALL    @RARGCV         ;GET RIGHT VALUE
7490 002506                                    CALL    @DYDTAB(R4)     ;CALL OP ROUTINE
7491 002512                                    CALL    @RESCVT         ;SAVE RESULT OF SCALAR OP
7492 002516  000406                            BR      3$              ;ERROR ON OVERFLOW
7493 002520                                    SOB     R0,2$           ;DO NEXT ELEMENT
7494
7495 002524  005337  003166'                    DEC     LARGLN          ;END OF LEFT?
7496 002530  001356                            BNE     DOUTER          ;NO! DO SOME MORE
7497 002532                                    RETURN                  ;All done
7498
7499 002534                            3$:     CALL    DYDEXP          ;EXPAND RESULT
7500 002540  163702  003160'                    SUB     LARGFL,R2       ;BACK UP LEFT 1 ELEMENT
7501 002544  163703  003202'                    SUB     RARGFL,R3       ;BACK UP RIGHT 1 ELEMENT
7502 002550  000752                            BR      1$              ;AND GO
7504 002552                                    LIST
7505
7506
7507
7508
7509
7510 002552                            MOPINS  PRDSET,0
7511                                   .SBTTL  PRDSET                          SETUP ROUTINE FOR INNER AND OUTER PRODUCTS
7512
7513                                   ;++
7514                                   ; FUNCTION DESCRIPTION:
7515                                   ;       This routine sets up the argument parameters and routines for
7516                                   ;       inner and outer product.  The top of the SI is assumed to point
7517                                   ;       to the Left argument.
7518                                   ; INPUTS:
7519                                   ;       R3 - Pointer to Right argument
7520                                   ;       R4 - (Right) Function number
7521                                   ;--
7522 002552                                    .PSECT  CODE
7523
7524 002552                            PRDSET::
7525 002552  052737  100000  004764'            BIS     #F.PRNT,FLAGS           ;Enable Printing
7526 002560  016502  177776                    MOV     SI.VAL(SI),R2           ;Get pointer to left argument
7527 002564  010237  003162'                    MOV     R2,LARGBL               ;Save pointer to Left argument block
7528 002570  011237  003164'                    MOV     (R2),LARGPT             ;And pointer to value
7529 002574  116200  000003                    MOVB    A.BITS(R2),R0           ;Get the Left argument's bits
7530 002600  002071                            BGE     2$                      ;Make sure it's an array
7531 002602  042700  177771                    BIC     #^C<A.NUM>,R0           ;Clear garbage bits
7532 002606  010037  003170'                    MOV     R0,LARGTY               ;Save Left argument's type
7533 002612  016037  000044'  003172'          MOV     SIZTAB(R0),LARGSF       ;Get Left argument's Shift count
7534 002620  016037  000054'  003174'          MOV     LENTAB(R0),LARGLE       ;Get Left argument's element size
7535 002626  016037  000054'  003160'          MOV     LENTAB(R0),LARGFL       ;Get Left argument's stride constant
```

PRDSET SETUP ROUTINE FOR INNER AND OUTER PRODUCTS

```
7536
7537 002634  010337  003204'                    MOV     R3,RARGBL               ;Save pointer to Right argument block
7538 002640  011337  003206'                    MOV     (R3),RARGPT             ;And pointer to value
7539 002644  116301  000003                     MOVB    A.BITS(R3),R1           ;Get Right argument's bits
7540 002650  002045                             BGE     2$                      ;Make sure it's an array
7541 002652  042701  177771                     BIC     #^C<A.NUM>,R1           ;Clear garbage bits
7542 002656  010137  003212'                    MOV     R1,RARGTY               ;Save Right argument's type
7543 002662  016137  000044' 003214'           MOV     SIZTAB(R1),RARGSF       ;Get Right argument's shift constant
7544 002670  016137  000054' 003216'           MOV     LENTAB(R1),RARGLE       ;Get Right argument's element size
7545 002676  016137  000054' 003202'           MOV     LENTAB(R1),RARGFL       ;Get Right argument's stride constant
7546 002704  010002                             MOV     R0,R2                   ;Assume Left type larger
7547 002706  020102                             CMP     R1,R2                   ;Is it?
7548 002710  003401                             BLE     1$                      ;No, Right type is
7549 002712  010102                             MOV     R1,R2                   ;Use Right type
7550 002714  006204                     1$:     ASR     R4                      ;Adjust function number to byte index
7551 002716  116400  000000G                   MOVB    DYDBTL(R4),R0           ;Get functions type bits
7552 002722  006304                             ASL     R4                      ;Restore function number
7553 002724                                     CALL    CMPTRT                  ;Compute (Right) function result type
7554 002730  010437  005116'                    MOV     R4,BLZBUB               ;Save the function number
7555 002734  010203                             MOV     R2,R3                   ;Copy intermediate input type
7556 002736  063702  003170'                    ADD     LARGTY,R2               ;Compute fetch table index
7557 002742  016237  000000G 003176'           MOV     LARGCT(R2),LARGCV       ;Get Left argument fetch routine
7558 002750  063703  003212'                    ADD     RARGTY,R3               ;Compute fetch table index
7559 002754  016337  000000G 003220'           MOV     RARGCT(R3),RARGCV       ;Get Right argument fetch routine
7560 002762                                     RETURN                          ;All setup now
7561
7562 002764  104402                     2$:     SYSERR                          ;Argument not an array block
7564 002766                                     LIST
7565
7566
7567
7568
7569
7570 002766                             MOPINS  INSPAC,0
7571                                     .SBTTL  INSPAC                  ALLOCATE SPACE FOR INNER PRODUCT
7572
7573                                     ;+
7574                                     ; FUNCTIONAL DESCRIPTION:
7575                                     ;       This routine allocates space for the result of the inner product
7576                                     ;       function
7577                                     ;-
7578 002766                             INSPAC::
7579 002766  116204  000002                        MOVB    A.DIMS(R2),R4           ;GET # OF DIMS ON LEFT
7580 002772                                        CONINC  R4                      ; FIXUP IF SCALAR
7581 002776  116301  000002                        MOVB    A.DIMS(R3),R1           ;GET # OF DIMS ON RIGHT
7582 003002                                        CONINC  R1                      ; FIXUP IF SCALAR
7583 003006  060401                                ADD     R4,R1                   ;GET TOTAL
7584 003010  162701  000002                        SUB     #2,R1                   ;MINUS TWO FOR COMBINING DIMENSIONS
7585 003014  022701  000020                        CMP     #DIMAX,R1               ;TOO MANY DIMS?
7586 003020  003001                                BGT     1$                      ;NOPE
7587 003022  104417                                DOMERR                          ;ARGS HAVE TOO MANY DIMS
7588 003024                             1$:        PUSH    <OPTMP5,R2,R3>          ;Save argument pointers
7589 003034  010403                                MOV     R4,R3                   ;Copy Left argument rank
7590 003036  005303                                DEC     R3                      ;Minus one for combined dimension
7591 003040  006303                                ASL     R3                      ;Get dimension list size in bytes
7592 003042  062703  000006                        ADD     #A.DIM1,R3              ;Add in header size
7593 003046  060203                                ADD     R2,R3                   ;Now add in Left argument pointer
```


INSPAC ALLOCATE SPACE FOR INNER PRODUCT

```
 7594 003050  010337  005100'                MOV     R3,OPTMP5               ;Save pointer to last dimension
 7595 003054  013703  005110'                MOV     OPTMP9,R3               ;GET LEFT ITERATION COUNT
 7596 003060                                 MUL     OPTM10,R3               ;MULTIPLY TO GET SIZE OF RESULT
 7597 003074                                 PUSH    <R3,R1>                 ;Save the Result size and rank
 7598 003100                                 CALL    GETSPA                  ;Get space for result and fill in its header
 7599 003104                                 POP     R4                      ;Get the Result size back
 7600 003106                                 POP     <R3,R2,OPTMP5>          ;Restore argument pointers
 7601 003116  116301  000002                MOVB    A.DIMS(R3),R1           ;GET # OF DIMS
 7602 003122                                 CONINC  R1                      ; FIXUP IF SCALAR
 7603 003126  062703  000010                ADD     #A.DIM1+2,R3            ;SKIP OVER FIRST DIM
 7604 003132  005301                        DEC     R1                      ;WE SKIPPED FIRST DIM
 7605 003134  003403                        BLE     3$                      ;RIGHT ARG IS VECTOR OR SCALAR
 7606
 7607 003136  012320              2$:       MOV     (R3)+,(R0)+             ;COPY DIM LIST FROM RIGHT PART
 7608 003140                                 SOB     R1,2$
 7609
 7610 003144              3$:       RETURN                                  ;Done
 7612 003146                                 LIST
```

B 2

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 89
INSPAC                  ALLOCATE SPACE FOR INNER PRODUCT

   7614 003146                          DOPINS  INNR,156
```

C 2

APL006 MACRO M1113 08-FEB-80 09:33 PAGE 90
INSPAC ALLOCATE SPACE FOR INNER PRODUCT

7960 003146 LIST

APL006 MACRO M1113 08-FEB-80 09:33 PAGE 91

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 91
INSPAC                  ALLOCATE SPACE FOR INNER PRODUCT

   7962 003146                          MOPINS  SUBR,0
```



```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 92
```

E 2

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 92
INSPAC                  ALLOCATE SPACE FOR INNER PRODUCT

   8725 003146                                  LIST
```

F 2

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 93
```

```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 93
INSPAC                    ALLOCATE SPACE FOR INNER PRODUCT

   8727 003146                          MOPINS  RAND,0
   8771 003146                                  LIST
```



```
APLO06  MACRO M1113  08-FEB-80 09:33  PAGE 94
```

G 2

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 94
INSPAC                  ALLOCATE SPACE FOR INNER PRODUCT

      1         000001                          .END
```

H 2

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 94-1
```

SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135          CS.ONE= 000100          D$SCON  001254RG  020 D.SUBS= 000006
AC1    =%000001          CHRLBR= 000073          CTEPS   000152R  025 D$SDIF  000516RG  020 D.TAKE= 000002
AC2    =%000002          CHRLPR= 000072          CTMAX   000212R  025 D$SUBS  001334RG  020 D.TRAN= 000004
AC3    =%000003          CHRMNS= 000137          CTRMAX  000172R  025 D$UNON  000302RG  020 D.UNON= 000006
AC4    =%000004          CHRMUL= 000075          C.ALOC= 000162          D.AND  = 000000          D.$MAX= 000000
AC5    =%000005          CHRNEG= 000100          C.BLOK= 000064          D.BXCT= 000007          D.$MIN= 000000
ALLINT  003274R     025  CHRPDL= 000271          C.BLSZ= 000140          D.BXIO= 000007          D.$MUL= 000000
ARYRET  003276R     025  CHRPLU= 000055          C.BUFF= 177760          D.BXNL= 000007          D.$PLU= 000000
ASCQUO= 000042           CHRPWR= 000120          C.BUFL= 177762          D.BXPP= 000007          D.$RES= 000000
A.ALEN= 000004           CHRQST= 000121          C.CCNT= 177766          D.BXPW= 000007          D.$SUB= 000000
A.ARG = 000040           CHRQUO= 000113          C.CPTR= 177764          D.BXRL= 000007          D.1COM= 000001
A.ASB = 000200           CHRRBR= 000047          C.CZ   = 001000         D.CIRC= 000001          D.1EXP= 000002
A.BITS= 000003           CHRRPR= 000042          C.DIG  = 000040         D.COMB= 000001          D.1ROT= 000003
A.CON = 000020           CHRSEM= 000074          C.FASW= 020000          D.COMP= 000001          EDICHR  005063R     025
A.DIMS= 000002           CHRSMO= 000112          C.FLEN= 000010          D.CONT= 000006          EDICNT  005064R     025
A.DIM1= 000006           CHRSPA= 000040          C.FLSH= 004000          D.CTLM= 000000          EDIFLG  005062R     025
A.NUM = 000006           CHRUPA= 000131          C.FRC  = 000001         D.DEAL= 000003          EDIT   = 000003
A.PTR = 000000           CHR.BS= 000010          C.IDEN= 000010          D.DECD= 000002          EIS    = 000000
A.TMP = 000010           CHR.CC= 000003          C.NAME= 000102          D.DIV  = 000000         ELESIZ  005132R     025
BADDIR= 104463           CHR.CR= 000015          C.NOBF= 000400          D.DMD  = 000010         ENDCOR  000004R     025
BADERR= 104437           CHR.CZ= 000032          C.NUM  = 000020         D.DQAD= 177777          EOF    = 104475
BLZBUB  005116R     025  CHR.GT= 000046          C.OP   = 000004         D.DROP= 000002          ERRMAX= 000166
BRSELF  005046R     025  CHR.LF= 000012          C.OPN  = 010000         D.ENCD= 000002          ERRNUM  004772R     025
CARVE = ****** GX        CHR.LT= 000043          C.RDLN= 177776          D.EQ   = 000000         ERRPTR  004770R     025
CCNT    005072R     025  CHR.SI= 000017          C.REWT= 002000          D.EXPD= 000002          ERR0    004774R     025
CHANEX  003060R     025  CHR.SL= 000057          C.RND  = 000010         D.FGET= 000005          EXEBUF  005054R     025
CHAN00  000256R     025  CHR.SO= 000016          C.SBIT= 177756          D.FMT  = 000004         E$CT    005056R     025
CHAN01  000330R     025  CHR.TB= 000011          C.SPFX= 000100          D.FPNT= 000005          E$PT    005060R     025
CHAN02  000512R     025  CHR.UL= 000106          C.SPGL= 000200          D.FPUT= 000005          FA.APD= 000100
CHAN03  000674R     025  CHR..A= 000141          C.SPLX= 000001          D.GE   = 000000         FA.CRE= 000010
CHAN04  001056R     025  CHR..E= 000145          C.SPRS= 000002          D.GT   = 000000         FA.DLK= 001000
CHAN05  001240R     025  CHR..S= 000163          C.SRC  = 177774         D.IBM  = 000002         FA.ENB= 100000
CHAN06  001422R     025  CHR..T= 000164          C.SRCP= 000024          D.IND  = 000002         FA.EXC= 002000
CHAN07  001604R     025  CHR..U= 000165          C.TT   = 000004         D.INNR= 000013          FA.EXT= 000004
CHAN08  001766R     025  CHR..Z= 000172          C.WIDC= 177770          D.INTR= 000006          FA.NSP= 000100
CHAN09  002150R     025  CHR..0= 000060          C.WIDR= 177772          D.LE   = 000000         FA.NSR= 000200
CHAN10  002332R     025  CHR..1= 000061          DEBUG  = 000000         D.LOG  = 000001         FA.POS= 010000
CHAN11  002514R     025  CH.AND= 000001          DEFERR= 104405          D.LT   = 000000         FA.RD  = 000001
CHAN12  002676R     025  CLROUT  000176R  020    DEVIL   005114R  025 D.MAX  = 000000         FA.RWD= 004000
CHAN13  000022R     023  CMDBUF  003250R  025    DEVNTF= 104470          D.MEM  = 000002         FA.SEQ= 040000
CHAN14  000204R     023  CMDMAX= 000010          DIGITS  000006R  025 D.MIN  = 000000         FA.SHR= 000040
CHNOPN= 104471           CMPTRT= ****** GX       DIMAX  = 000020         D.MUL  = 000000         FA.TMP= 000020
CHRAND= 000051           CNTSIZ  005130R  025    DIMBLK  003112R  025 D.NE   = 000000         FA.WCK= 020000
CHRBEL= 000007           COMBIN  005076R  025    DIMPTR  003114R  025 D.NN   = 000000         FA.WRT= 000002
CHRBKR= 000133           COMCNT  005126R  025    DIMTYP  003116R  025 D.NR   = 000000         FD.BLK= 000010
CHRBOL= 000200           COMCT  = ****** GX      DIRFUL= 104466          D.OR   = 000000         FD.CCL= 000002
CHRBOX= 000114           COMERR= 104425          DOMERR= 104417          D.PLUS= 000000         FD.COM= 020000
CHRCMA= 000054           COMPTR  005122R  025    DONE    000274R  020 D.PSUB= 000006         FD.CR  = 000002
CHRCOL= 000076           COMRTN  003156R  025    DOUTER  002466RG  020 D.PWR  = 000001         FD.DIR= 000010
CHRDEL= 000107           COMSKP  005124R  025    DPTERR= 104416          D.QQAD= 000000         FD.FTN= 000001
CHRDIV= 000053           CONTNS  001270R  020    DYDBTL= ****** GX       D.QUAD= 000000         FD.F11= 040000
CHRDLR= 000176           CONTN1  001302R  020    DYDCNT  003224R  025 D.RES  = 000000         FD.INS= 000010
CHRDLT= 000110           CORPTR  005120R  025    DYDEXP= ****** GX       D.RHO  = 000000         FD.ISP= 002000
CHRDOT= 000056           CS.ASS= 000200          DYDTAB= ****** GX       D.ROT  = 000003         FD.MNT= 100000
CHREP  = 000105          CS.BOO= 000010          D$CONT  001264RG  020 D.SCON= 000006         FD.OSP= 004000
CHRERR= 104431           CS.FLO= 000020          D$INTR  000442RG  020 D.SDIF= 000006         FD.PLC= 000004
CHRFIL= 000177           CS.IDN= 000040          D$PSUB  001324RG  020 D.SUB  = 000000         FD.PRN= 000004
```

SYMBOL TABLE

```
FD.PSE= 010000          F.BR  = 000020          F.STBK= 000036          LCOUNT  005106R     025 M.FIL = 000177
FD.RAH= 000001          F.CHR = 000075          F.STK = 000022          LENERR= 104412          M.FLR = 177777
FD.RAN= 000002          F.CNTG= 000034          F.STOP= 000004          LENTAB  000054R     025 M.FMT = 000004
FD.REC= 000001          F.CODE= 000010          F.STP = 000001          LEXPTR  003100R     025 M.GDUP= 000003
FD.RWM= 000001          F.DFNB= 000046          F.SUSP= 000002          LFTOP   005066R     025 M.GDWN= 000003
FD.SDI= 000020          F.DSPT= 000044          F.TRC = 000002          LINC    005102R     025 M.GETI= 000006
FD.SQD= 000040          F.DVNM= 000134          F.UNIT= 000136          LINE    005016R     025 M.IBM = 000003
FD.TTY= 000004          F.EFBK= 000010          F.URBD= 000020          LININC  005026R     025 M.IDN = 000000
FD.WBH= 000002          F.EFN = 000050          F.VAL = 177776          LISIZE= 000020          M.INSP= 000006
FF.CHR= 000005          F.END = 002000          F.VBN = 000064          LITOP   005072R     025 M.IOTA= 000000
FF.NV = 000003          F.EOBB= 000032          F.VBSZ= 000060          LNLNG = 104541          M.LIT = 000205
FF.PVE= 000002          F.ERR = 000052          GETFRA= ****** GX       LOCLCT  005036R     025 M.LN  = 000001
FF.FWD= 000001          F.FACC= 000043          GETIDN  002012RG    020 LOWCOR  000002R     025 M.MMD = 000010
FF.RWF= 000006          F.FFBY= 000014          GETL  = ****** GX       LSTERP  004766R     025 M.NEG = 000000
FF.SPC= 000004          F.FLAG= 000002          GETLIN= ****** GX       LWIDTH  000070R     025 M.NOT = 000000
FHEADR  005012R     025 F.FLB = 000100          GETREG= ****** GX       L.BITS= 000003          M.NUB = 000006
FHRET   005040R     025 F.FNAM= 000110          GETSPA  002310RG    020 L.CNT = 000002          M.NUM = 000204
FILNTF= 104467          F.FNB = 000102          GNOME   003102R     025 L.PRNT= 000001          M.OUTR= 000012
FIS   = 000000          F.FNDF= 000010          HASHC   005136R     025 L.PTR = 000000          M.PITM= 000001
FISCON= 046000          F.FTYP= 000116          HASHTB  005140R     025 L.PTR1= 000004          M.PRDS= 0000C
FIXPTR  003074R     025 F.FUN = 000010          HSHMSK= 177601          L.SUBS= 000002          M.QQAD= 00000C
FLAGS   004764R     025 F.FVER= 000120          HSHSIZ= 000200          MACHIN= 000106          M.QUAD= 000000
FLCBUF  003250R     025 F.HEAD= 000006          IASH  = ****** GX       MAXCHN= 000014          M.RAND= 000012
FLINE   005014R     025 F.HIBK= 000004          ICHAN   005050R     025 MONBLK  003120R     025 M.RAVL= 000000
FLOEXP  003270R     025 F.LBL = 000020          ILFNAM= 104464          MONCVT  003132R     025 M.RCIP= 000000
FLONEG  003272R     025 F.LINE= 000012          ILLCHN= 104540          MONLEN  003130R     025 M.RDCT= 000011
FLSWTB  000222R     025 F.LOC = 000004          ILLSW = 104565          MONPTR  003122R     025 M.REVS= 000003
FLTLEN= 000010          F.LSTC= 000004          IMUL  = ****** GX       MONSFT  003126R     025 M.ROLL= 000000
FNINSI  005044R     025 F.LUN = 000042          INDERR= 104410          MONTYP  003124R     025 M.SATA= 000001
FORIGN  000032R     025 F.MBCT= 000054          INNTMP  003250R     025 MONVAL  003134R     025 M.SCAN= 000011
FO.APD= 000106          F.MBC1= 000055          INNTM2  005016R     025 M$DQAD  001660RG    020 M.SHAP= 000000
FO.MFY= 000002          F.MBFG= 000056          INPTR   000064R     025 M$NUB   000000RG    020 M.SIGN= 000000
FO.RD = 000001          F.MON = 000002          INSNAM= 104462          M.ABS = 000000          M.SUBR= 000001
FO.UPD= 000006          F.NIL = 000001          INSPAC  002766RG    020 M.BOL = 000200          M.SYM = 000203
FO.WRT= 000016          F.NRBD= 000024          IOERR = 104477          M.BXAV= 000007          M.SYN = 000201
FPMODE  000042R     025 F.NREC= 000030          I.CNT = 177776          M.BXCR= 000007          M.TRAN= 000012
FPP   = 000000          F.OPN = 000010          I.DIM = 177772          M.BXCT= 000007          M.$ABS= 00C000
FPPLEN= 000004          F.OVBS= 000030          I.LEN = 000012          M.BXEX= 000007          M.$CEI= 000000
FREBLK  007400R     025 F.PREV= 000000          I.OERR= 000062          M.BXFX= 000007          M.$FLR= 000000
FUZZ    000142R     025 F.PRNT= 100000          I.OLD = 177770          M.BXIO= 000007          M.$IDN= 000000
FXPTR1  003076R     025 F.PROT= 010000          I.PTR = 177774          M.BXLC= 000003          M.$NEG= 000000
F$EVAL= 000100          F.RACC= 000016          I.STR = 177766          M.BXNC= 000007          M.$SIG= 000000
F$EVI = 000200          F.RATT= 000001          LABERR= 104406          M.BXNL= 000007          M.1RDT= 000011
F$EXEC= 000400          F.RCNM= 000034          LABLPT  005042R     025 M.BXPP= 000007          M.1REV= 000003
F$FUNC= 001000          F.RCTL= 000017          LARGBL  003162R     025 M.BXPW= 000007          M.1SCN= 000011
F.ACTL= 000076          F.READ= 004000          LARGCT= ****** GX       M.BXRL= 000007          NAMBUF  003240R     025
F.ALOC= 000040          F.REDF= 000040          LARGCV  003176R     025 M.BXTS= 000007          NB.DEV= 000200
F.BBFS= 000062          F.RES = 000040          LARGFL  003160R     025 M.BXWA= 000007          NB.DIR= 000100
F.BDB = 000070          F.RET = 177772          LARGLE  003174R     025 M.CEI = 177777          NB.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002          LARGLN  003166R     025 M.CHR = 000202          NB.SD1= 000400
F.BITS= 177774          F.RTYP= 000000          LARGPT  003164R     025 M.CMT = 000206          NB.SD2= 001000
F.BKDN= 000026          F.R2  = 000014          LARGSF  003172R     025 M.DOUT= 000006          NB.SNM= 000040
F.BKDS= 000020          F.R3  = 000016          LARGTY  003170R     025 M.DQAD= 000006          NB.STP= 000020
F.BKEF= 000050          F.R4  = 000020          LARGVA  003200R     025 M.ENCD= 000004          NB.SVR= 000010
F.BKP1= 000051          F.SEQN= 000100          LBINC   005100R     025 M.EXE = 000000          NB.TYP= 000002
F.BKST= 000024          F.SPDV= 000072          LBUFF   003316R     025 M.EXP = 000001          NB.VER= 000001
F.BKVB= 000064          F.SPUN= 000074          LBUFLN= 000621          M.FACT= 000001          ND.DEV= 000200
```

SYMBOL TABLE

```
ND.EXT= 000002          OPTMP0  005066R 025 RESVAL  003154R 025 SUBSET  001340R 020 TTYLEN= 000622
ND.NAM= 000004          OPTMP1  005070R 025 RINC    005104R 025 SUBST1  001350R 020 TURKEY= 104436
ND.PPN= 000100          OPTMP2  005072R 025 RLTBAS  005066R 025 SUBST2  001354R 020 TYPE  = 000166
ND.$EX= 000000          OPTMP3  005074R 025 ROTINC  005070R 025 SYMB1   005344R 025 T.BOL  = 020000
ND.$E$= 000020          OPTMP4  005076R 025 ROUND   000010R 025 SYMF    005340R 025 T.CON  = 004000
ND.$NM= 000000          OPTMP5  005100R 025 ROVER   003312R 025 SYMPT1  005346R 025 T.DIM  = 000100
ND.$N$= 000040          OPTMP6  005102R 025 RSTS  = 000000      SYMTAB  005352R 025 T.FUN  = 010000
ND.$PN= 001400          OPTMP7  005104R 025 RSX   = 000115      SYMTBF= 104404      T.LIST= 000400
NEGSGN  005063R 025     OPTMP8  005106R 025 RTOP    005070R 025 SYMTOP  005350R 025 T.LIT  = 000020
NINPT1  003302R 025     OPTMP9  005110R 025 RT11  = 000000      SYMTP   006272R 025 T.MON  = 000010
NINPT2  003300R 025     OPTM10  005112R 025 R.BITS= 000003      SYMWRN= 000054      T.NIL  = 000004
NINSTK  003304R 025     OPTM14  0050[illegible] 025 R.FIX = 000001      SYNERR= 104407      T.NUM  = 002000
NINTMP  003310R 025     OP.CEI= ****** GX       R.LCT = 000002      SYSERR= 104402      T.OPFN= 000001
NINTYP  003306R 025     OP.FLR= ****** GX       R.LN  = 000002      SYSTAT  000000R 024 T.OTHR= 040000
NOTAVL= 104472          ORIGIN  000030R 025 R.LNUM= 000010      SY.BAD= 000001      T.SOP  = 000002
NOTOPN= 104473          OUTPTR  000066R 025 R.NEXT= 000006      SY.BRK= 100000      T.SPOP= 000040
NPRT1   004762R 025     OVROPR= 000006          R.OFF = 000000      SY.BS  = 000010      T.TMP  = 000200
NUB1    000100R 020     PATCH   007344R 025 R.PREV= 000004      SY.CLR= 000040      T.VAL  = 100000
NULL    007364R 025     PRDSET  002552RG 020 R.PTR = 000000      SY.ERP= 000020      T.VAR  = 001000
NUMCHN= 000020          PRECSN  000052R 025 R.SEQ = 000003      SY.NOU= 000100      UPPER  = 177400
NUMDFL  003230R 025     RANDAC  003064R 025 R.VAR = 000002      SY.OPN= 000002      UTNAME  000344R 023
NUMEFL  003233R 025     RANDY   003062R 025 R6    =%000006      SY.PMT= 001000      VALERR= 104413
NUMESN  003236R 025     RANERR= 104411          R7    =%000007      SY.SRT= 000400      VARBLE  007400R 025
NUMFLG  003230R 025     RARGBL  003204R 025 SETOP   000544R 020 SY.SVC= 000004      VARBLS  000000R 025
NUMLEN  000062R 025     RARGCT= ****** GX       SETOP1  000606R 020 SY.SVL= 000200      VERNUM  000000R 025
NUMNEG  003232R 025     RARGCV  003220R 025 SETOP2  000632R 020 S.AB  = 177776      VRSION= 000002
NUMQFL  003234R 025     RARGFL  003202R 025 SETOP3  000660R 020 S.ASB = 100000      WIDTH   000072R 025
NUMRIG  003231R 025     RARGLE  003216R 025 SETOP4  000704R 020 S.BFHD= 000020      WROUND  000020R 025
NUMWID  003226R 025     RARGLN  003210R 025 SETOP5  000712R 020 S.BITS= 000002      WSDATE  000136R 025
NUMZFL  003235R 025     RARGPT  003206R 025 SETUP   000740R 020 S.CHR1= 000003      WSFULL= 104403
NYIERR= 104432          RARGSF  003214R 025 SI    =%000005      S.FATT= 000016      WSNAME  000074R 025
N.BLSZ= 000036          RARGTY  003212R 025 SIDAMN= 104433      S.FDB = 000140      WSTIME  000132R 025
N.DESC= 000020          RARGVA  003222R 025 SIFRAM  003314R 025 S.FLB = 040000      $FIS   = ****** G
N.DEV = 000032          RBASE   005074R 025 SISTAK  006350R 025 S.FNAM= 000006      $FPP   = ****** G
N.DEVN= 000034          RCOUNR  005112R 025 SITOP   006346R 025 S.FNB = 000036      $FPPLN= ****** G
N.DID = 000024          RCOUNT  005110R 025 SITP    007144R 025 S.FNBW= 000017      $MACHN= ****** G
N.DVNM= 000032          REDCNT  005070R 025 SIWARN= 000200      S.FNTY= 000004      $RSTS  = ****** G
N.EXT = 000014          REDIGN  005066R 025 SIZTAB  000044R 025 S.FTYP= 000002      $RSX   = ****** G
N.FID = 000000          REDOFF  005072R 025 SI.BIT= 177774      S.LEN = 000002      $RT11  = ****** G
N.FNAM= 000006          REENTR  003060R 025 SI.LEN= 177772      S.LINK= 177774      $STF   = ****** GX
N.FTYP= 000014          REGTYP= 000003          SI.RET= 177772      S.NFEN= 000020      .BOOLE= 000002
N.FVER= 000016          RESBIT  003142R 025 SI.VAL= 177776      S.TYPE= 140000      .CHARA= 000000
N.NAME= 000006          RESCT = ****** GX       SKPSIZ  005134R 025 S.VAL = 000000      .DEF.  = 000002
N.NEXT= 000022          RESCVT  003152R 025 SPHOLD  000004R 024 TKDIMS  005072R 025 .FLOAT= 000006
N.PPN = 000024          RESERR= 104440          SPINIT  000002R 024 TKFRAM  005066R 025 .FSRCB= ****** G
N.STAT= 000020          RESLEN  003150R 025 STKINC= 000774      TKLEFT  005070R 025 .INTEG= 000004
N.UNIT= 000034          RESPTR  003140R 025 STOP  = 104420      TMODE   000403R 023 .MOLUN  000404R 023
OBUFLN= 001000          RESSFT  003146R 025 SUBCNT  003104R 025 TMPTAG= 000166          ...GBL= 000000
OCHAN   005052R 025     RESTYP  003144R 025 SUBDCT  003106R 025 TRNTAB= ****** GX       ...TPC= 000020
OMODE   000402R 023     RESULT  003136R 025 SUBPTR  003110R 025 TTYBUF  004140R 025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
```

K 2

```
APL006  MACRO M1113  08-FEB-80 09:33  PAGE 94-4
SYMBOL TABLE

MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    003146      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13624 WORDS  ( 54 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:45
APL006,DM1:[305,301]APL006/-SP=APLRX1,APLDEF,APL006,APLOP,END
```

L 2

```
**FILE**ID**APL007
```

FILEID**APL007

```
  AAAAAA      PPPPPPPP      LL              000000        000000     77777777
  AAAAAA      PPPPPPPP      LL              000000        000000     77777777
AA      AA    PP      PP    LL            00    00    00    00           77
AA      AA    PP      PP    LL            00    00    00    00           77
AA      AA    PP      PP    LL            00    00    00    0000         77
AA      AA    PP      PP    LL            00    00    00    0000         77
AA      AA    PPPPPPPP      LL            00    00    00  00  00       77
AA      AA    PPPPPPPP      LL            00    00    00  00  00       77
AAAAAAAAAA    PP            LL            00    00    0000    00     77
AAAAAAAAAA    PP            LL            00    00    0000    00     77
AA      AA    PP            LL            00    00    00      00   77
AA      AA    PP            LL            00    00    00      00   77
AA      AA    PP            LLLLLLLLLL      000000        000000   77
AA      AA    PP            LLLLLLLLLL      000000        000000   77

LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL              SSSSSS        TT
LL              SSSSSS        TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LLLLLLLLLL    SSSSSSSS        TT
LLLLLLLLLL    SSSSSSSS        TT
```

....
....
....
....

.MAIN. MACRO M1113 08-FEB-80 09:37
TABLE OF CONTENTS

.MAIN. MACRO M1113 08-FEB-80 09:37
TABLE OF CONTENTS

2- 1 APLDEF
2- 8 A P L
2- 9
2- 10 A PROGRAMMING LANGUAGE
2- 11
2- 12 STEVE POULSEN & DAVE VANN
2- 13 11-Jul-76
2- 14 ROBERT GOTTLIEB & PETER HAYNES
2- 15 2-Feb-79
2- 16
2- 17 V2
2- 18
2- 19
2- 20 COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
2- 21 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
2- 22
2- 23
2- 24 THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
2- 25 COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
2- 26 COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
2- 27 BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
2- 28 USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
2- 29 TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
2- 30 DEC.
2- 31
2- 32 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
2- 33 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
2- 34 CORPORATION.
2- 35
2- 36 DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
2- 37 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
2- 38
2- 39
3- 46 DEFINITIONS ASSEMBLY PARAMETERS
4- 139 DEFINITIONS OPERATOR OVERLAYS
5- 268 DEFINITIONS REGISTERS
6- 331 DEFINITIONS MEMORY ORGANIZATION
7- 377 DEFINITIONS ORDERING PSECTS
8- 426 DEFINITIONS SI PARSE FRAMES
9- 474 DEFINITIONS TOKEN DESCRIPTION BITS
10- 515 DEFINITIONS SI EXECUTION FRAMES
11- 580 DEFINITIONS SPECIAL PARSE CHARACTERS
12- 607 DEFINITIONS CHRTAB BITS
12- 624 DEFINITIONS SYSTEM STATUS BITS
13- 640 DEFINITIONS F-FRAMES
14- 724 DEFINITIONS FUNCTION LINE BLOCKS
16- 807 DEFINITIONS ARRAY STORAGE BLOCK
17- 869 DEFINITIONS LIST STORAGE BLOCK
18- 911 DEFINITIONS INDEXING STACK FRAME
19- 940 DEFINITIONS SYMBOL TABLE ENTRY
20- 969 DEFINITIONS CHANNEL BLOCK
22- 1085 DEFINITIONS RSX DATA
23- 1110 DEFINITIONS NAME BLOCKS
33- 1612 DEFINITIONS MACROS
35- 1905 DEFINITIONS APL CHARACTERS
36- 1967 DEFINITIONS ERRORS
76- 5202 SYSTEM VARIABLE ATOMIC VECTOR

APL007 MACRO M1113 08-FEB-80 09:37
TABLE OF CONTENTS

APL007 MACRO M1113 08-FEB-80 09:37
TABLE OF CONTENTS

76- 5226	SYSTEM VARIABLE	COMPARISON TOLERANCE OUTPUT
76- 5238	SYSTEM VARIABLE	COMPARISON TOLERANCE INPUT
76- 5279	SYSTEM VARIABLE	INDEX ORIGIN INPUT
76- 5291	SYSTEM VARIABLE	INDEX ORIGIN OUTPUT
76- 5348	SYSTEM VARIABLE	PRINT PRECISION INPUT
76- 5360	SYSTEM VARIABLE	PRINT PRECISION OUTPUT
76- 5389	SYSTEM VARIABLE	PRINT WIDTH OUTPUT
76- 5401	SYSTEM VARIABLE	PRINT WIDTH INPUT
76- 5424	SYSTEM VARIABLE	RANDOM LINK INPUT
76- 5436	SYSTEM VARIABLE	RANDOM LINK OUTPUT
76- 5454	SYSTEM VARIABLE	TIME STAMP
76- 5570	SYSTEM VARIABLE	WORKSPACE AVAILABLE
77- 5595	SYSTEM FUNCTION	CANONICAL REPRESENTATION
78- 5719	SYSTEM FUNCTION	EXPUNGE
79- 5778	SYSTEM FUNCTION	FIX FUNCTION
80- 5882	SYSTEM FUNCTION	NAME CLASSIFI ATION
81- 5953	SYSTEM FUNCTION	NAME LIST
81- 5967	SYSTEM FUNCTION	NAME LIST

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 1


      1        000000                  RT11    =  0            ;NOT RT-11
      2        000000                  RSTS    =  0            ;NOT RSTS/E
      3        000115                  RSX     = 'M            ;RSX-11/M
      4
      5        000000                  EIS     =  0            ;NO EIS
      6        000000                  FIS     =  0            ;NO FIS
      7        000000                  FPP     =  0            ;NO FPP
      8
      9        000004                  FPPLEN  =  4.           ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 09:37  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
APL002  MACRO M1113  08-FEB-80 09:37  PAGE 44
```

E 3

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  APLOO7
                                .IDENT  \V02.03\

                                ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
                                ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
                                ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
                                ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
                                ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLOO7 IS:
                                ;
                                ;       APLOO7 < ...,APLOO7,APLOP
                                ;
                                ;

000007                          OVROPR= 7                       ;OVERLAY SPECIFIED OPERATORS
```

F 3

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 46
```

```
      5
      6                                 ;++
      7                                 ; FUNCTIONAL DESCRIPTION:
      8                                 ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
      9                                 ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
     10                                 ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
     11                                 ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
     12                                 ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
     13                                 ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
     14                                 ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
     15                                 ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
     16                                 ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
     17                                 ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
     18                                 ;       INTACT!
     19                                 ;--
     24
     27
    124
    125                                 ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
    126                                 ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
    127                                 ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
    128                                 ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
    129                                 ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
    130                                 ;MACROS.
    131
    132                                 .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
    133                                 .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
    134                                 .IF EQ OVROPR-D.'NAME
    135                                         .PSECT  CODE
    136                                 D$'NAME::
    137                                 .ENDM
    138
    139                                 .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
    140                                 .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
    141                                 .IF EQ OVROPR-M.'NAME
    142                                         .PSECT  CODE
    143                                         .IF NE NUM
    144                                 M$'NAME::
    145                                     .ENDC
    146                                 .ENDM
    147
    148                                 .MACRO  LIST
    149                                         .ENDC
    150                                 .ENDM
    151
    152                                 .MACRO  PRMRES
    153                                         .IF EQ 1
    154                                 .ENDM
    155
    159                                                 ;*********** ASSEMBLING THE FILE!!!!!
    160 007400                          PRMRES
    199 007400                                  LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,,FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
618
619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
639 007400                                  LIST
```

```
    641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
    652 007400                                  LIST
    653
    654
    655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
    662 007400                                  LIST
    663
    664
    665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
    676 007400                                  LIST
    677
    678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
    687 007400                                  LIST
    688
    689
    690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
    700 007400                                  LIST
    701
    702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    712 007400                                  LIST
    713
    714
    715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    733 007400                                  LIST
    734
    735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
    745 007400                                  LIST
    746
    747
    748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
    758 007400                                  LIST
    759
    760
    761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
    780 007400                                  LIST
    781
```

```
783 007400                              MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
786 007400                                      LIST
787
788 007400                              MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
791 007400                                      LIST
792
793 007400                              MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
799 007400                                      LIST
800
801
802 007400                              MOPINS  $IDN,-2
808 007400                                      LIST
809
810
811 007400                              MOPINS  $FLR,-14
855 007400                                      LIST
856
857
858 007400                              MOPINS  $CEI,-16
868 007400                                      LIST
869
870
871 007400                              MOPINS  NEG,4,T.SOP,,INTEGER
878 007400                                      LIST
879
880 007400                              MOPINS  $NEG,-4
887 007400                                      LIST
888
889
890 007400                              MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
901 007400                                      LIST
902
903 007400                              MOPINS  $SIG,-6
914 007400                                      LIST
915
916
917 007400                              MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
928 007400                                      LIST
929
930 007400                              MOPINS  EXP,20,T.SOP,,FLOAT
939 007400                                      LIST
940
941
942 007400                              MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
949 007400                                      LIST
950
951
952 007400                              MOPINS  PITM,52,T.SOP,,FLOAT
959 007400                                      LIST
960
961 007400                              PRMRES
966 007400                                      LIST
967
968
969 007400                              MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
977 007400                                      LIST
978
```

```
  979 007400                          MOPINS  $ABS,-12
  992 007400                                  LIST
  993
  994
  995 007400                          MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
 1002 007400                                  LIST
 1003
 1004
 1005 007400                          MOPINS  NOT,50
 1031 007400                                  LIST
 1032
 1033
 1034 007400                          MOPINS  ROLL,100
 1057 007400                                  LIST
```

DEFINITIONS ERRORS

```
   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

DEFINITIONS ERRORS

```
1189 007400                          MOPINS  SHAP,102
1211 007400                                  LIST
1212
1213
1214
1215
1216
1217 007400                          DOPINS  RHO,102
1313 007400                                  LIST
```

DEFINITIONS ERRORS

```
   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

B 4

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

C 4

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 54
DEFINITIONS     ERRORS

   1701 007400                          DOPINS  1COM,124
   1709 007400                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 007400                          DOPINS  COMP,60,T.DIM
   1789 007400                                  LIST
```

D 4

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

E 4

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 54
```

E 4

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

F 4

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 57
DEFINITIONS     ERRORS

   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```



```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 58
```

DEFINITIONS ERRORS

```
2206 007400                          DOPINS  1ROT,146
2214 007400                                  LIST
2215
2216
2217
2218
2219
2220 007400                          DOPINS  ROT,144,T.DIM
2328 007400                                  LIST
```

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```



```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 60
```

```
APLO07  MACRO M1113  08-FEB-80 09:37  PAGE 60
DEFINITIONS     ERRORS

   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```



```
APLO07  MACRO M1113  08-FEB-80 09:37  PAGE 61
```

DEFINITIONS ERRORS

```
   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 63
DEFINITIONS     ERRORS

   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```



```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 64
```

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 64
DEFINITIONS     ERRORS

   3221 007400                          MOPINS  MMD,122
   3316 007400                                  LIST
```



```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 65
```

DEFINITIONS ERRORS

```
   3318 007400                          DOPINS  DMD,122
```

DEFINITIONS ERRORS

3864 007400 LIST

DEFINITIONS ERRORS

```
   3866 007400                          DOPINS  MEM,70
   3921 007400                                  LIST
```

D 5


```
 3923 007400                            MOPINS  NUB,106
 3986 007400                                    LIST
 3987
 3988
 3989
 3990
 3991
 3992 007400                            DOPINS  UNON,106
 4022 007400                                    LIST
 4023
 4024
 4025
 4026
 4027
 4028 007400                            DOPINS  INTR,66
 4042 007400                                    LIST
 4043
 4044
 4045
 4046
 4047
 4048 007400                            DOPINS  SDIF,50
 4161 007400                                    LIST
```

E 5

E 5

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 69
DEFINITIONS     ERRORS

   4163 007400                          DOPINS  SCON,234
   4179 007400                                  LIST
   4180
   4181
   4182 007400                          DOPINS  CONT,112
   4200 007400                                  LIST
   4201
   4202
   4203 007400                          DOPINS  PSUB,236
   4209 007400                                  LIST
   4210
   4211
   4212 007400                          DOPINS  SUBS,116
   4288 007400                                  LIST
```

F 5

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 70
DEFINITIONS     ERRORS

   4290 007400                          MOPINS  EXE,70
   4336 007400                                  LIST
```

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 71
DEFINITIONS     ERRORS

   4338 007400                          MOPINS  ENCD,76
   4411 007400                                  LIST
```

APL007 MACRO M1113 08-FEB-80 09:37 PAGE 72
DEFINITIONS   ERRORS

```
4413 007400                             MOPINS  FMT,160
4423 007400                                     LIST
4424
4425
4426
4427
4428
4429 007400                             DOPINS  FMT,160
```


APL007 MACRO M1113 08-FEB-80 09:37 PAGE 73

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 73
DEFINITIONS     ERRORS

   4953 007400                                  LIST
```



```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 74
```

DEFINITIONS ERROPS

```
4955 007400                          MOPINS  IBM,132,,,MONIBM
4958 007400                                  LIST
4959
4960
4961 007400                          DOPINS  IBM,132
5046 007400                                  LIST
```

```
5048 007400                         MOPINS  QUAD,74,T.NIL+T.SPOP
5075 007400                                 LIST
5076
5077
5078
5079
5080
5081 007400                         DOPINS  QUAD,74,T.NIL+T.SPOP
5097 007400                                 LIST
5098
5099
5100
5101
5102
5103 007400                         MOPINS  QQAD,142,T.NIL+T.SPOP
5135 007400                                 LIST
5136
5137
5138
5139
5140
5141 007400                         DOPINS  QQAD,142,T.NIL+T.SPOP
5152 007400                                 LIST
5153
5154
5155
5156
5157
5158 007400                         MOPINS  DQAD,134,T.NIL+T.SPOP
5192 007400                                 LIST
5193
5194
5195
5196 007400                         DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 007400                                 LIST
```

```
5201 007400                            MOPINS  BXAV,162,T.NIL
5202                                   .SBTTL  SYSTEM VARIABLE                 ATOMIC VECTOR
5203
5204 000000  012700  000410                    MOV     #<A.DIM1+2>+400,R0 ;LENGTH OF RESULT
5205 000004                                    CALL    GETCOR             ;GET MEMORY
5206 000010                                    SIFRAM  ,#T.VAL            ;Build an SI frame
5207 000022  010010                            MOV     R0,@R0             ;POINT TO SELF
5208 000024  062720  000010                    ADD     #A.DIM1+2,(R0)+ ;POINT TO DATA (TO BE)
5209 000030  012720  100001                    MOV     #<A.ASB+.CHARACTER>*400+1,(R0)+ ;CHARACTER VECTOR
5210 000034  012701  000400                    MOV     #400,R1            ;LENGTH
5211 000040  010120                            MOV     R1,(R0)+           ;ALEN
5212 000042  010120                            MOV     R1,(R0)+           ;DIM 1
5213 000044  005002                            CLR     R2                 ;CLEAR CHAR VAL
5214 000046  110220                    1$:     MOVB    R2,(R0)+           ;MOVE CHAR TO VECTOR
5215 000050  005202                            INC     R2                 ;INCREMENT
5216 000052                                    SOB     R1,1$              ;LOOP
5217 000056                                    RETURN
5219 000060                                    LIST
5220
5221
5222
5223
5224
5225 000060                            MOPINS  BXCT,164,T.NIL+T.SPOP
5226                                   .SBTTL  SYSTEM VARIABLE                 COMPARISON TOLERANCE OUTPUT
5227
5228 000060  012704  000000'                    MOV     #BXCT,R4           ;POINT TO DESCRIPTOR
5229 000064                                    CALLR   SAME               ;COPY/RETURN
5230
5231 000000                            .PSECT  TABLES
5232 000000  000142' 107000  000001    BXCT::  .WORD   FUZZ,<A.ASB+.FLOAT+A.TMP>*400,1,1
     000006  000001
5234 000010                                    LIST
5235
5236
5237 000010                            DOPINS  BXCT,164,T.NIL+T.SPOP
5238                                   .SBTTL  SYSTEM VARIABLE                 COMPARISON TOLERANCE INPUT
5239
5240 000070  012701  000006                    MOV     #.FLOAT,R1         ;WANT FLOAT VALUE
5241 000074                                    CALL    GETSCA             ;CHECK LENGTH, CONVERT
5242 000100                                    LDF     @(R0),AC0          ;GET THE VALUE
5243 000110                                    CFCC                       ;CHECK CONDITION CODES
5244 000114  002475                            BLT     1$                 ;THIS NUMBER IS TOO SMALL
5245 000116                                    CMPF    CTMAX,AC0          ;CHECK HOW BIG IT IS
5246 000126                                    CFCC                       ;CHECK CONDITION CODES
5247 000132  002466                            BLT     1$                 ;THIS NUMBER IS TOO BIG
5248 000134                                    STF     AC0,FUZZ           ;THIS NUMBER IS JUST RIGHT
5249 000144                                    PUSH    <R0,R1>            ;Get some working registers
5250 000150  012700  000010                    MOV     #8.,R0             ;Table element size
5251 000154  012701  000152'                   MOV     #CTEPS,R1          ;Point to constant table
5252 000160                                    STF     AC0,(R1)           ;EPS constant for positive arguments = Fuzz
5253 000166  060001                            ADD     R0,R1              ;Next table entry
5254 000170                                    LDF     #^F1.0,AC3         ;Get a floating point one
5255 000200                                    SUBF    AC0,AC3            ;1 - BXCT
5256 000206                                    STF     AC3,AC2            ;Copy it
5257 000214                                    DIVF    AC3,AC0            ;BXCT / (1 - BXCT)
5258 000222                                    STF     AC0,(R1)           ;EPS constant for minus args = BXCT/(1-BXCT)
```

```
5259 000230  060001                          ADD     R0,R1           ;Next table entry
5260 000232                                   LDF     #^F1.0,AC0      ;Get a floating point one again
5261 000242                                   ADDF    AC0,AC2         ;2 - BXCT
5262 000250                                   DIVF    AC2,AC0         ;1 / (2 - BXCT)
5263 000256                                   STF     AC0,(R1)        ;RMAX constant for plus args = 1/(2-BXCT)
5264 000264  060001                          ADD     R0,R1           ;Next table entry
5265 000266                                   DIVF    AC2,AC3         ;(1 - BXCT) / (2 - BXCT)
5266 000274                                   STF     AC3,(R1)        ;RMAX constant for minus args =(1-BXCT)/(2-BXCT)
5267 000302                                   POP     <R1,R0>         ;Restore working registers
5268 000306                                   RETURN                  ;FINE'
5269
5270 000310  104417                  1$:     DOMERR                  ;WRONGO
5272 000312                                   LIST
5273
5274
5275
5276
5277
5278 000312                          MOPINS   BXIO,166,T.NIL+T.SPOP
5279                                 .SBTTL   SYSTEM VARIABLE         INDEX ORIGIN INPUT
5280
5281 000312  012704  000010'                 MOV     #BXIO,R4        ;POINT TO FRAME
5282 000316                                   CALLR   SAME            ;COPY/RETURN
5283
5284 000010                          .PSECT   TABLES
5285 000010  000030' 105000  000001  BXIO::  .WORD   ORIGIN,<A.ASB+.BOOLEAN+A.TMP>*400,1,1
     000016  000001
5287 000020                                   LIST
5288
5289
5290 000020                          DOPINS   BXIO,166,T.NIL+T.SPOP
5291                                 .SBTTL   SYSTEM VARIABLE         INDEX ORIGIN OUTPUT
5292
5293 000322  012701  000002                 MOV     #.BOOLEAN,R1    ;CONVERT TO BOOLEAN
5294 000326                                   CALL    GETSCA          ;CHECK SIZE/CONVERT
5295 000332  117000  000000                 MOVB    @(R0),R0        ;GET VALUE
5296 000336  042700  177776                 BIC     #^C<1>,R0       ;CLEAR HIGH GARBAGE
5297 000342  010037  000030'                 MOV     R0,ORIGIN       ;STORE ORIGIN
5298 000346                                   LDCIF   R0,AC0          ;CONVERT TO FLOAT
5299 000354                                   STF     AC0,FORIGN      ;SAVE FLOATING ORIGIN
5300 000364                                   RETURN
5302 000366                                   LIST
5303
5304
5305
5306
5307 000366                          MOPINS   BXLC,212,T.NIL
5340 000366                                   LIST
5341
5342
5343
5344
5345
5346
5347 000366                          MOPINS   BXPP,170,T.NIL+T.SPOP
5348                                 .SBTTL   SYSTEM VARIABLE         PRINT PRECISION INPUT
5349
```

```
5350 000366  012704  000020'                    MOV     #BXPP,R4          ;POINT TO FRAME
5351 000372                                     CALLR   SAME              ;COPY/RETURN
5352
5353 000020                             .PSECT  TABLES
5354 000020  000006' 106000  000001    BXPP::  .WORD   DIGITS,<A.ASB+.INTEGER+A.TMP>*400,1,1
     000026  000001
5356 000030                                     LIST
5357
5358
5359 000030                             DOPINS  BXPP,170,T.NIL+T.SPOP
5360                                    .SBTTL  SYSTEM VARIABLE           PRINT PRECISION OUTPUT
5361
5362 000376  012701  000004                    MOV     #.INTEGER,R1      ;CONVERT TO INTEGER
5363 000402                                     CALL    GETSCA            ;CHECK LENGTH/CONVERT
5364 000406  017000  000000                    MOV     @(R0),R0          ;GET THE VALUE
5365 000412  003433                             BLE     3$                ;TOO SMALL, ERROR
5366 000414  012701  000007                    MOV     #7,R1             ;MAX ALLOWED IS 7 FOR SINGLE PREC
5367 000420  022737  000010  000062'           CMP     #8.,NUMLEN        ;ARE WE DOUBLE PRECISION?
5368 000426  001002                             BNE     1$                ;NO, MUST BE SINGLE
5369 000430  062701  000012                    ADD     #10.,R1           ;MAX IS NOW 17.
5370 000434  020001                    1$:     CMP     R0,R1             ;HOW BIG IS IT, JOHNNIE?
5371 000436  003021                             BGT     3$                ;IT'S SO BIG THAT...
5372 000440  010037  000006'                    MOV     R0,DIGITS         ;STORE # OF DIGITS
5373 000444                                     LDF     #^F5.0,AC0        ;COMPUTE ROUND
5374 000454                             2$:     DIVF    #^F10.0,AC0       ;DIVIDE BY 10.
5375 000464                                     SOB     R0,2$             ;LOOP FOR ALL DIGITS
5376 000470                                     STF     AC0,ROUND         ;SAVE ROUNDING CONSTANT
5377 000500                                     RETURN                    ;RETURN
5378
5379 000502  104417                     3$:     DOMERR                    ;NOT IN VALID RANGE
5381 000504                                     LIST
5382
5383
5384
5385
5386
5387
5388 000504                             MOPINS  BXPW,172,T.NIL+T.SPOP
5389                                    .SBTTL  SYSTEM VARIABLE           PRINT WIDTH OUTPUT
5390
5391 000504  012704  000030'                    MOV     #BXPW,R4          ;POINT TO FRAME
5392 000510                                     CALLR   SAME              ;COPY/RETURN
5393
5394 000030                             .PSECT  TABLES
5395 000030  000072' 106000  000001    BXPW::  .WORD   WIDTH,<A.ASB+.INTEGER+A.TMP>*400,1,1
     000036  000001
5397 000040                                     LIST
5398
5399
5400 000040                             DOPINS  BXPW,172,T.NIL+T.SPOP
5401                                    .SBTTL  SYSTEM VARIABLE           PRINT WIDTH INPUT
5402
5403 000514  012701  000004                    MOV     #.INTEGER,R1      ;CONVERT TO INTEGER
5404 000520                                     CALL    GETSCA            ;CHECK LENGTH/CONVERT
5405 000524  017000  000000                    MOV     @(R0),R0          ;GET NEW WIDTH
5406 000530  022700  000036                    CMP     #^D30,R0          ;IS IT TOO SMALL?
5407 000534  003012                             BGT     1$                ;YES
```

```
5408 000536 022700 000621                  CMP     #LBUFLN,R0          ;IS IT TOO BIG?
5409 000542 002407                         BLT     1$                  ;YES
5410 000544 010037 000072'                 MOV     R0,WIDTH            ;AHHH... JUST RIGHT
5411 000550 110037 000250'                 MOVB    R0,CHAN00+C.WIDR    ;TTY RESET WIDTH
5412 000554 110037 000246'                 MOVB    R0,CHAN00+C.WIDC    ;TTY CURRENT WIDTH
5413 000560                                RETURN                      ;FINISHED
5414
5415 000562 104417                 1$:     DOMERR                      ;***ERROR***
5417 000564                                LIST
5418
5419
5420
5421
5422
5423 000564                        MOPINS  BXRL,174,T.NIL+T.SPOP
5424                               .SBTTL  SYSTEM VARIABLE             RANDOM LINK INPUT
5425
5426 000564 012704 000040'                 MOV     #BXRL,R4            ;POINT TO FRAME
5427 000570                                CALLR   SAME                ;COPY/RETURN
5428
5429 000040                                .PSECT  TABLES
5430 000040 003062' 106000 000001  BXRL::  .WORD   RANDY,<A.ASB+.INTEGER+A.TMP>*400,1,1
     000046 000001
5432 000050                                LIST
5433
5434
5435 000050                        DOPINS  BXRL,174,T.NIL+T.SPOP
5436                               .SBTTL  SYSTEM VARIABLE             RANDOM LINK OUTPUT
5437
5438 000574 012701 000004                  MOV     #.INTEGER,R1        ;CONVERT TO INTEGER
5439 000600                                CALL    GETSCA              ;CHECK LENGTH/CONVERT
5440 000604 017000 000000                  MOV     @(R0),R0            ;GET THE VALUE
5441 000610 003403                         BLE     1$                  ;TOO SMALL, ERROR
5442 000612 010037 003062'                 MOV     R0,RANDY            ;STORE NEW RANDOM SEED
5443 000616                                RETURN                      ;RETURN
5444
5445 000620 104417                 1$:     DOMERR                      ;NOT IN VALID RANGE
5447 000622                                LIST
5448
5449
5450
5451
5452
5453 000622                        MOPINS  BXTS,200,T.NIL
5454                               .SBTTL  SYSTEM VARIABLE             TIME STAMP
5455
5457 000622 012700 000030                  MOV     #A.DIM1+18.,R0      ;Total size of hacked Time Stamp
5461 000626                                CALL    GETFRA              ;Get arithmentic frame
5462 000632 062720 000010                  ADD     #A.DIM1+2.,(R0)+    ;Point to data
5463 000636 012720 102001                  MOV     #<A.ASB+.INTEGER>*400+1,(R0)+
5464                                                                   ;Integer Vector
5465 000642 012720 000007                  MOV     #7.,(R0)+           ;Length 7. words
5466 000646 012720 000007                  MOV     #7.,(R0)+           ;Also 1st dim
5467 000652 010002                         MOV     R0,R2               ;Save for later
5469 000654                                GTIM$S  R0                  ;Get the time
     000654 010046                         MOV     R0,-(SP)
     000656 012746                         MOV     (PC)+,-(SP)
```

```
        000660     075     002                  .BYTE   61.,2
        000662  104377                          EMT     ^O<377>
   5470 000664  062712  003554                  ADD     #1900.,(R2)     ;Time has centuries, too
   5471 000670  016200  000014                  MOV     12.(R2),R0      ;Get the tick
   5472 000674                                  MUL     #1000.,R0       ;Multiply for Millisec
   5473 000710                                  DIV     14.(R2),R0      ;Divide by ticks/Sec
   5474 000724  010062  000014                  MOV     R0,12.(R2)      ;Put in the Millisecond
   5554 000730                                  RETURN                  ;Finished
   5555
   5562 000732                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 000732                          MOPINS  BXWA,176,T.NIL
   5570                                 .SBTTL  SYSTEM VARIABLE         WORKSPACE AVAILABLE
   5571
   5572 000732  012700  000010                  MOV     #<A.DIM1+2>,R0  ;Space for array block header
   5573 000736  063700  000062'                 ADD     NUMLEN,R0       ;Its only a floating point scalar
   5574 000742                                  CALL    GETFRA          ;Get result, SI frame for it
   5575 000746  062720  000010                  ADD     #A.DIM1+2,(R0)+ ;  Pointer to the data
   5576 000752  012720  103000                  MOV     #<A.ASB+.FLOAT>*400,(R0)+  ;Bits and Rank
   5577 000756  012720  000001                  MOV     #1,(R0)+        ;  ALEN
   5578 000762  012720  000001                  MOV     #1,(R0)+        ;  DIM1
   5579 000766                                  CALL    SCAMEM          ;Now Scan memory for free space
   5587 000772                                  LDCIF   R1,AC0          ;Convert number of free words to floating
   5588 001000                                  ADDF    AC0,AC0         ;Get it in bytes
   5589 001006                                  STF     AC0,(R0)        ;Put it into Result
   5590 001014                                  RETURN
   5592 001016                                  LIST
```

```
5594 001016                            MOPINS  BXCR,206
5595                                   .SBTTL  SYSTEM FUNCTION          CANONICAL REPRESENTATION
5596
5597 001016  013701  003120'                    MOV     MONBLK,R1                ;Point to data block
5598 001022  132761  000006  000003             BITB    #A.NUM,A.BITS(R1)        ;Is it character?
5599 001030  001044                             BNE     3$                       ;No, DOMAIN ERROR
5600 001032  122761  000001  000002             CMPB    #1.,A.DIMS(R1)           ;Vector or scalar?
5601 001040  002441                             BLT     4$                       ;No, RANK ERROR
5602 001042  016103  000004                     MOV     A.ALEN(R1),R3            ;Get the character count
5603 001046  011104                             MOV     (R1),R4                  ;   and the data pointer
5604 001050                                     GETC    2$                       ;Get the 1st character
     001050  005303                             DEC R3                           ;ANY CHARACTERS LEFT?
     001052  002430                             BLT 2$                   ;NO, BRANCH TO 2$
     001054  112402                             MOVB (R4)+,R2                    ;YES, GET NEXT CHAR
5605 001056  042702  177400                     BIC     #<UPPER>,R2              ;Clear the upper bits
5606 001062  132762  000010  000000G           BITB    #C.IDEN,CHRTAB(R2)       ;Is it an identifier?
5607 001070  001421                             BEQ     2$                       ;Nope. is illegal name
5608 001072                                     CALL    SYMBOL                   ;Is it in the symbol table?
5609 001076  005703                             TST     R3                       ;Any chars left?
5610 001100  002403                             BLT     1$                       ;Nope
5611 001102                                     CALL    GETNB2                   ;Any non-blank characters left?
5612 001106  002012                             BGE     2$                       ;Yes
5613 001110  011001                     1$:     MOV     (R0),R1                  ;Point to the item
5614 001112  001410                             BEQ     2$                       ;Nothing there
5615 001114  032761  040000  000002             BIT     #S.FLB,2.(R1)            ;Is it a Function?
5616 001122  001404                             BEQ     2$                       ;Nope
5617 001124  132761  000004  000003             BITB    #F.LOC,R.BITS(R1)        ;Is it locked?
5618 001132  001405                             BEQ     5$                       ;No, is ok
5619 001134  005003                     2$:     CLR     R3                       ;No lines here
5620 001136  005004                             CLR     R4                       ;None at all
5621 001140  000476                             BR      9$                       ;Go
5622
5623 001142  104417                     3$:     DOMERR                           ;Domain error
5624
5625 001144  104411                     4$:     RANERR                           ;Rank Error
5626                                    ;+
5627                                    ; After all that checking, we now know that we have an APL
5628                                    ; function that we can convert to Canonical Form. Do it!
5629                                    ;-
5630 001146  010137  005012'            5$:     MOV     R1,FHEADR                ;Save the pointer to the header
5631 001152  012702  003060'                    MOV     #CHANEX,R2               ;Point to the execute channel
5632 001156                                     PUSH    OCHAN                    ;Save output channel
5633 001162  010237  005052'                    MOV     R2,OCHAN                 ;Use Execute Channel
5634 001166  012700  000011                     MOV     #<-C.SBIT>/2,R0          ;Loop counter
5635 001172                             6$:     PUSH    -(R2)                    ;Push execute channel info
5636 001174                                     SOB     R0,6$                    ;Loop for all
5637 001200  012702  003060'                    MOV     #CHANEX,R2               ;Pointer to Execute channel
5638 001204  012700  001004                     MOV     #516.,R0                 ;Temporary space for lines
5639 001210                                     CALL    GETFRA                   ;Get a frame, put it on the SI
5640 001214  062720  000004                     ADD     #4.,(R0)+                ;Point to data space
5641 001220  012720  100000                     MOV     #S.ASB,(R0)+             ;Indicate array
5642 001224  012762  010000  177756             MOV     #C.OPN,C.SBIT(R2)        ;Set the BITS field
5643 001232  010062  177760                     MOV     R0,C.BUFF(R2)            ;C.BUFF
5644 001236  012762  001000  177762             MOV     #512.,C.BUFL(R2)         ;C.BUFL
5645 001244  005003                             CLR     R3                       ;Maximum line width
5646 001246  005004                             CLR     R4                       ;Count of # of lines
5647 001250  016262  177760  177764     7$:     MOV     C.BUFF(R2),C.CPTR(R2)    ;Copy C.BUFF to C.CPTR
```

APLOO7 MACRO M1113 08-FEB-80 09:37 PAGE 77-1
SYSTEM FUNCTION CANONICAL REPRESENTATION

```
5648 001256  016262  177762  177766          MOV     C.BUFL(R2),C.CCNT(R2)   ;Copy C.BUFL to C.CCNT
5649 001264  005062  177772                  CLR     C.WIDR(R2)              ;Clear C.WIDR
5650 001270  005062  177770                  CLR     C.WIDC(R2)              ;  and C.WIDC
5651 001274                                  CALL    FLNLST                  ;List the line
5652 001300  016200  177764                  MOV     C.CPTR(R2),R0           ;Get the Data pointer
5653 001304  166200  177760                  SUB     C.BUFF(R2),R0           ;   minus the start pointer
5654 001310  020003                          CMP     R0,R3                   ;Largest line yet?
5655 001312  002401                          BLT     8$                      ;Nope
5656 001314  010003                          MOV     R0,R3                   ;Yes, ge' new large size
5657 001316  005204                  8$:     INC     R4                      ;Increment line count
5658 001320  016101  000006                  MOV     R.NEXT(R1),R1           ;Point to the next line
5659 001324  020137  005012'                 CMP     R1,FHEADR               ;Is it the beginning?
5660 001330  001347                          BNE     7$                      ;No, continue to loop
5661                                 ;+
5662                                 ; Now that we know the # of lines and the maximum line width,
5663                                 ; we will delete the temporary space, compute the amount of space needed,
5664                                 ; and comtinue
5665                                 ;-
5666 001332                          9$:     CALL    KILL                    ;Delete the temporary space
5667 001336                                  PUSH    R3                      ;Save the max line width
5668 001340                                  MUL     R4,R3                   ;Compute the space needed
5669 001352  103505                          BCS     15$                     ;Overflow, error
5670 001354  010300                          MOV     R3,R0                   ;Copy the amount
5671 001356  062700  000012                  ADD     #A.DIM1+4.,R0           ;Add the amount of overhead
5672 001362                                  CALL    GETCOI                  ;Ask for the space
5673 001366  000477                          BR      15$                     ;Sorry, we ain't got that much
5674 001370                                  SIFRAM  ,#T.VAL                 ;Standard frame for character
5675 001402  010010                          MOV     R0,(R0)                 ;Point to self
5676 001404  062720  000012                  ADD     #A.DIM1+4.,(R0)+        ;Point to beginning of data
5677 001410  012720  100002                  MOV     #<A.ASB+.CHARACTER>*400+2.,(R0)+
5678                                                                         ;Character, 2 dimensions
5679 001414  010320                          MOV     R3,(R0)+                ;A.ALEN
5680 001416                                  POP     R3                      ;Get back the max line width
5681 001420  010420                          MOV     R4,(R0)+                ;# of lines
5682 001422  010320                          MOV     R3,(R0)+                ;Maximum line width
5683 001424  005760  177772                  TST     -6.(R0)                 ;Anything there?
5684 001430  001455                          BEQ     17$                     ;Nothing there, finished
5685 001432  016062  177772  177762          MOV     -6.(R0),C.BUFL(R2)      ;Get the buffer size
5686 001440  010062  177760                  MOV     R0,C.BUFF(R2)           ;Set the buffer pointer
5687 001444  016262  177760  177764  10$:    MOV     C.BUFF(R2),C.CPTR(R2)   ;Copy C.BUFF to C.CPTR
5688 001452  016262  177762  177766          MOV     C.BUFL(R2),C.CCNT(R2)   ;Copy C.BUFL to C.CCNT
5689 001460  005062  177772                  CLR     C.WIDR(R2)              ;Clear C.WIDR
5690 001464  005062  177770                  CLR     C.WIDC(R2)              ;  and C.WIDC
5691 001470                                  CALL    FLNLST                  ;List the line
5692 001474  060362  177760                  ADD     R3,C.BUFF(R2)           ;Increment buffer pointer
5693 001500  160362  177762                  SUB     R3,C.BUFL(R2)           ;   and decrement count
5694 001504  016200  177764                  MOV     C.CPTR(R2),R0           ;Copy the pointer
5695 001510  166262  177760  177764          SUB     C.BUFF(R2),C.CPTR(R2)   ;Compute -<Fill Count>
5696 001516  002005                          BGE     12$                     ;None needed
5697 001520  112720  000040          11$:    MOVB    #CHRSPA,(R0)+           ;Copy in fill
5698 001524  005262  177764                  INC     C.CPTR(R2)              ;Loop until
5699 001530  002773                          BLT     11$                     ;Finished
5700 001532  016101  000006          12$:    MOV     R.NEXT(R1),R1           ;Point to the next line
5701 001536                                  SOB     R4,10$                  ;Loop for all the lines
5702 001542  062702  177756          13$:    ADD     #C.SBIT,R2              ;Point to start for execute channel
5703 001546  012700  000011                  MOV     #<-C.SBIT>/2,R0         ;Loop count
5704 001552                          14$:    POP     (R2)+                   ;Restore the execute
```

```
 5705 001554                                        SOB     R0,14$                  ;   channel
 5706 001560                                        POP     OCHAN                   ;Restore the output channel
 5707 001564                                17$:    RETURN                          ;Finished!
 5708                                               ;+
 5709                                               ; Error handling
 5710                                               ;-
 5711 001566  012766  001576' 000024        15$:    MOV     #16$,2.-C.SBIT(SP)      ;Change the return address
 5712 001574  000762                                BR      13$                     ;   and go
 5713
 5714 001576  104403                        16$:    WSFULL                          ;Workspace full
 5716 001600                                        LIST
```

```
5718 001600                                MOPINS  BXEX,202
5719                                       .SBTTL  SYSTEM FUNCTION          EXPUNGE
5720
5721 001600  013702  003120'                 MOV     MONBLK,R2               ;Point to data block
5722 001604  132762  000006  000003          BITB    #A.NUM,A.BITS(R2)       ;Is it character?
5723 001612  001113                          BNE     7$                      ;No, is DOMAIN ERROR
5724 001614  016201  000006                  MOV     A.DIM1(R2),R1           ;Get length of vector
5725 001620  012700  000001                  MOV     #1.,R0                  ;(We are assuming a vector)
5726 001624  122762  000002  000002          CMPB    #2.,A.DIMS(R2)          ;Well, is it one?
5727 001632  003005                          BGT     1$                      ;Yes, it's true, it's true
5728 001634  002503                          BLT     8$                      ;No, is RANK ERROR
5729 001636  016200  000006                  MOV     A.DIM1(R2),R0           ;Get the # of names to scan
5730 001642  016201  000010                  MOV     A.DIM1+2(R2),R1         ;    and the length of each line
5731 001646                          1$:     PUSH    <R1,R0>                 ;Save the values for the loop
5732 001652  012737  000000G 003152'         MOV     #SM.BB,RESCVT           ;Save the store convert routine
5733 001660  062700  000010                  ADD     #A.DIM1+2.,R0           ;Add in overhead
5734 001664                                  CALL    GETFRA                  ;Build the frame
5735 001670  062720  000010                  ADD     #A.DIM1+2.,(R0)+        ;Point to data area
5736 001674  116220  000002                  MOVB    A.DIMS(R2),(R0)+        ;Move in the result size
5737 001700  001402                          BEQ     2$                      ;Special if source is scalar
5738 001702  105360  177777                  DECB    -1.(R0)                 ;Otherwise reduce rank by 1
5739 001706  112720  000202          2$:     MOVB    #A.ASB+.BOOLEAN,(R0)+   ;Specify the result type
5740 001712  011620                          MOV     (SP),(R0)+              ;A.ALEN
5741 001714  011620                          MOV     (SP),(R0)+              ;A.DIM1
5742 001716  001446                          BEQ     6$                      ;Nothing there, already finished
5743 001720  010001                          MOV     R0,R1                   ;Copy pointer into result pointer
5744 001722                                  PUSH    MONPTR                  ;Get pointer to source data
5745
5746 001726  016603  000004          3$:     MOV     4.(SP),R3               ;Get size of a line
5747 001732  011604                          MOV     (SP),R4                 ;Pointer to next string element
5748 001734  060316                          ADD     R3,(SP)                 ;Increment pointer
5749 001736  005037  003154'                 CLR     RESVAL                  ;Assume we'll fail
5750 001742                                  GETC    5$                      ;Get the 1st character
     001742  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
     001744  002425                          BLT 5$                  ;NO, BRANCH TO 5$
     001746  112402                          MOVB (R4)+,R2           ;YES, GET NEXT CHAR
5751 001750  042702  177400                  BIC     #<UPPER>,R2             ;Clear out garbage bits
5752 001754  132762  000010  000000G         BITB    #C.IDEN,CHRTAB(R2)      ;Is it a valid character?
5753 001762  001416                          BEQ     5$                      ;No, must be an error
5754 001764                                  PUSH    R1                      ;Save result pointer
5755 001766                                  CALL    SYMBOL                  ;Lookup the symbol name
5756 001772                                  POP     R1                      ;Restore pointer
5757 001774  005703                          TST     R3                      ;ANYTHING LEFT?
5758 001776  002403                          BLT     4$                      ;NOPE
5759 002000                                  CALL    GETNB2                  ;Any non-blank characters left?
5760 002004  002005                          BGE     5$                      ;Yes, error
5761 002006                          4$:     CALL    ERASE                   ;Erase (Expunge) the item
5762 002012  001002                          BNE     5$                      ;It appears to have failed
5763 002014  005237  003154'                 INC     RESVAL                  ;Indicate success
5764 002020                          5$:     CALL    @RESCVT                 ;Store the result
5765 002024  104402                          SYSERR                          ;Should never happen
5766 002026  005366  000002                  DEC     2.(SP)                  ;Decrement loop count
5767 002032  003335                          BGT     3$                      ;Loop until exhausted
5768
5769 002034  062706  000006          6$:     ADD     #6.,SP                  ;Pop garbage from the stack
5770 002040                                  RETURN                          ;Finished
5771
```

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 78-1
SYSTEM FUNCTION          EXPUNGE

   5772 002042  104417                    7$:     DOMERR
   5773 002044  104411                    8$:     RANERR
   5775 002046                                    LIST
```

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 79
SYSTEM FUNCTION          EXPUNGE

 5777 002046                                MOPINS  BXFX,210
 5778                                        .SBTTL  SYSTEM FUNCTION          FIX FUNCTION
 5779
 5780 002046  013702  003120'                       MOV     MONBLK,R2                ;Point to data block
 5781 002052  132762  000006  000003               BITB    #<A.NUM>,A.BITS(R2)      ;Is it numeric?
 5782 002060  001052                               BNE     1$                       ;Yes, DOMAIN ERROR
 5783 002062  122762  000002  000002               CMPB    #2.,A.DIMS(R2)           ;Must have rank = 2
 5784 002070  001047                               BNE     2$                       ;Nope, RANK ERROR
 5785 002072  012700  000012                       MOV     #A.DIM1+4.,R0            ;Integer Scalar
 5786 002076                                       CALL    GETFRA                   ;Build an SI Frame
 5787 002102  062720  000010                       ADD     #A.DIM1+2.,(R0)+         ;Add in offset
 5788 002106  012720  102000                       MOV     #<A.ASB+.INTEGER>*400,(R0)+
 5789                                                                               ;Integer Scalar
 5790 002112  012720  000001                       MOV     #1,(R0)+                 ;A.ALEN
 5791 002116  012720  000001                       MOV     #1,(R0)+                 ;A.DIM1
 5792 002122  013710  000030'                      MOV     ORIGIN,(R0)              ;The Line with the error is
 5793 002126  010037  005072'                      MOV     R0,OPTMP2                ;Save address for increment
 5794                                        ;+
 5795                                        ; Set-up finished, now we scan the header line
 5796                                        ;-
 5797 002132  016237  000006  005074'               MOV     A.DIM1(R2),OPTMP3        ;Get the # of lines
 5798 002140  003424                               BLE     3$                       ;Too Small, Error
 5799 002142  016201  000010                       MOV     A.DIM1+2(R2),R1          ;Get the length of a line
 5800 002146  003421                               BLE     3$                       ;Too Small, Error
 5801 002150  010137  005076'                      MOV     R1,OPTMP4                ;Save it for later
 5802 002154  011200                               MOV     (R2),R0                  ;Pointer to data
 5803 002156  010037  005100'                      MOV     R0,OPTMP5                ;Save this, too, for later
 5804 002162  005037  005044'                      CLR     FNINSI                   ;Assume not on SI
 5805 002166  005037  005012'                      CLR     FHEADR                   ;New definition
 5806 002172  005037  005016'                      CLR     LINE                     ;Clear the line number
 5807 002176                                       CALL    FNHEAD                   ;Parse the header line
 5808 002202  000404                               BR      4$                       ;Error - is pre-defined or something
 5809 002204  000426                               BR      5$                       ;Not defined, OK
 5810
 5811 002206  104417                        1$:    DOMERR                           ;Domain Error
 5812 002210  104411                        2$:    RANERR                           ;Rank Error
 5813 002212  104412                        3$:    LENERR                           ;Length Error
 5814                                        ;+
 5815                                        ; Value is pre-defined, check for function
 5816                                        ;-
 5817 002214  103513                        4$:    BCS     11$                      ;Bad definition error
 5818 002216  011200                               MOV     (R2),R0                  ;Point to the pointer to the name
 5819 002220  032760  040000  000002               BIT     #S.FLB,2.(R0)            ;Is the old thing a Function?
 5820 002226  001506                               BEQ     11$                      ;No
 5821 002230  010200                               MOV     R2,R0                    ;Pointer to pointer to ...
 5822 002232                                       CALL    ERASE                    ;Attempt to erase
 5823 002236  001102                               BNE     11$                      ;Failed, give up
 5824 002240  013701  005076'                      MOV     OPTMP4,R1                ;Length of line
 5825 002244  013700  005100'                      MOV     OPTMP5,R0                ;Pointer to line
 5826 002250  005037  005012'                      CLR     FHEADR                   ;No header
 5827 002254                                       CALL    FNHEAD                   ;Try again
 5828 002260  104402                               SYSERR                           ;Abismal failure
 5829                                        ;+
 5830                                        ; It worked - now we have the header block
 5831                                        ;-
 5832 002262  010437  005012'               5$:    MOV     R4,FHEADR                ;Save Pointer to Function Header
 5833 002266  005337  005074'               6$:    DEC     OPTMP3                   ;Decrement loop count
```



```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 79-1
SYSTEM FUNCTION          EXPUNGE
```

```
APLO07  MACRO M1113  08-FEB-80 09:37  PAGE 79-1
SYSTEM FUNCTION          FIX FUNCTION

 5834 002272  003431                        BLE     9$                      ;Finished
 5835 002274  005277  005072'                INC     @OPTMP2                 ;Increment line # for error
 5836 002300  005237  005016'                INC     LINE                    ;Increment Line Number
 5837 002304  013703  005076'                MOV     OPTMP4,R3               ;Get the line length
 5838 002310  060337  005100'                ADD     R3,OPTMP5               ;Increment pointer to line
 5839 002314  013704  005100'                MOV     OPTMP5,R4               ;Get new pointer to the line
 5840 002320  010402                         MOV     R4,R2                   ;Copy line pointer
 5841 002322  060302                         ADD     R3,R2                   ;Point to end of line
 5842 002324  124227  000040          7$:    CMPB    -(R2),#CHRSPA           ;A Space?
 5843 002330  001003                         BNE     8$                      ;Nope
 5844 002332                                 SOB     R3,7$                   ;Loop for each character
 5845 002336  005203                         INC     R3                      ;Just in case, at least 1 character
 5846 002340                          8$:    GETC    13$                     ;Get a character
      002340  005303                         DEC R3                          ;ANY CHARACTERS LEFT?
      002342  002446                         BLT 13$                 ;NO, BRANCH TO 13$
      002344  112402                         MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
 5847 002346                                 CALL    FNLINE                  ;Scan the line
 5848 002352  000435                         BR      12$                     ;Error - bad syntax or something
 5849 002354  000744                         BR      6$                      ;Loop for all it's worth
 5850                                 ;+
 5851                                 ; Finished, now we clean up and go home
 5852                                 ;-
 5853 002356                          9$:    CALL    KILL                    ;Kill the numeric value
 5854 002362  017702  005012'                MOV     @FHEADR,R2              ;Pointer to the function name
 5855 002366  116204  000002                MOVB    S.LEN(R2),R4            ;Get the name length (Bytes)
 5856 002372  010400                         MOV     R4,R0                   ;Copy for allocation
 5857 002374  062700  000010                ADD     #A.DIM1+2,R0            ;Size of vector
 5858 002400                                 CALL    GETFRA                  ;Get an array
 5859 002404  042765  002000  177774        BIC     #T.NUM,SI.BITS(SI)      ;Clear the Numeric bit
 5860 002412  062720  000010                ADD     #A.DIM1+2,(R0)+         ;Increment the array pointer
 5861 002416  012720  100001                MOV     #<A.ASB+.CHARACTER>*400+1,(R0)+
 5862                                                                        ;A.BITS/A.DIMS
 5863 002422  010420                         MOV     R4,(R0)+                ;# of elements
 5864 002424  010420                         MOV     R4,(R0)+                ;Length of vector
 5865 002426  062702  000003                ADD     #S.CHR1,R2              ;Pointer to name
 5866 002432  112220                  10$:   MOVB    (R2)+,(R0)+             ;Copy the name
 5867 002434                                 SOB     R4,10$                  ;Character by character
 5868 002440                                 CALL    FNEND                   ;End the function
 5869 002444                          11$:   RETURN                          ;Finished
 5870                                 ;+
 5871                                 ; Error - Kill the new stuff
 5872                                 ;-
 5873 002446  017700  005012'         12$:   MOV     @FHEADR,R0              ;Pointer to name block
 5874 002452                                 CALL    ERASE                   ;Die die die!
 5875 002456  001772                         BEQ     11$                     ;And return
 5876
 5877 002460  104402                  13$:   SYSERR                          ;Error, finished
 5879 002462                                 LIST
```

```
5881 002462                                MOPINS  BXNC,204
5882                                       .SBTTL  SYSTEM FUNCTION          NAME CLASSIFICATION
5883
5884 002462  013702  003120'                       MOV     MONBLK,R2               ;Point to data block
5885 002466  132762  000006  000003               BITB    #A.NUM,A.BITS(R2)       ;Is it character?
5886 002474  001141                               BNE     7$                      ;No, is DOMAIN ERROR
5887 002476  016201  000006                       MOV     A.DIM1(R2),R1           ;Get length of vector
5888 002502  012700  000001                       MOV     #1.,R0                  ;(We are assuming a vector)
5889 002506  122762  000002  000002               CMPB    #2.,A.DIMS(R2)          ;Well, is it one?
5890 002514  003005                               BGT     1$                      ;Yes, it's true, it's true
5891 002516  003131                               BGT     8$                      ;No, is RANK ERROR
5892 002520  016200  000006                       MOV     A.DIM1(R2),R0           ;Get the # of names to scan
5893 002524  016201  000010                       MOV     A.DIM1+2(R2),R1         ;   and the length of each line
5894 002530                                1$:     PUSH    <R1,R0>                 ;Save the values for the loop
5895 002534  012737  000000G 003152'               MOV     #SM.II,RESCVT           ;Save the store convert routine
5896 002542  006300                               ASL     R0                      ;Integer result
5897 002544  062700  000010                       ADD     #A.DIM1+2.,R0           ;Add in overhead
5898 002550                                       CALL    GETFRA                  ;Build the frame
5899 002554  062720  000010                       ADD     #A.DIM1+2.,(R0)+        ;Point to data area
5900 002560  116220  000002                       MOVB    A.DIMS(R2),(R0)+        ;Move in the result size
5901 002564  001402                               BEQ     2$                      ;Special if source is scalar
5902 002566  105360  177777                       DECB    -1.(R0)                 ;Otherwise reduce rank by 1
5903 002572  112720  000204                2$:     MOVB    #A.ASB+.INTEGER,(R0)+   ;Specify the result type
5904 002576  011620                               MOV     (SP),(R0)+              ;A.ALEN
5905 002600  011620                               MOV     (SP),(R0)+              ;A.DIM1
5906 002602  001473                               BEQ     6$                      ;Nothing there, already finished
5907 002604  010001                               MOV     R0,R1                   ;Copy pointer into result pointer
5908 002606                                       PUSH    MONPTR                  ;Get pointer to source data
5909
5910 002612  016603  000004                3$:     MOV     4.(SP),R3               ;Get size of a line
5911 002616  011604                               MOV     (SP),R4                 ;Pointer to data
5912 002620  060316                               ADD     R3,(SP)                 ;Increment pointer
5913 002622  012737  000004  003154'               MOV     #4.,RESVAL              ;Assume we'll fail
5914 002630                                       GETC    5$                      ;Get the 1st character
     002630  005303                               DEC R3                          ;ANY CHARACTERS LEFT?
     002632  002451                               BLT 5$                  ;NO, BRANCH TO 5$
     002634  112402                               MOVB (R4)+,R2           ;YES, GET NEXT CHAR
5915 002636  042702  177400                       BIC     #<UPPER>,R2             ;Clear out garbage bits
5916 002642  132762  000010  000000G               BITB    #C.IDEN,CHRTAB(R2)      ;Is it a valid character?
5917 002650  001442                               BEQ     5$                      ;No, must be an error
5918 002652                                       PUSH    R1                      ;Save result pointer
5919 002654                                       CALL    SYMBOL                  ;Lookup the symbol name
5920 002660                                       POP     R1                      ;Restore pointer
5921 002662  005703                               TST     R3                      ;ANYTHING LEFT?
5922 002664  002403                               BLT     4$                      ;NOPE
5923 002666                                       CALL    GETNB2                  ;Any non-blank characters left?
5924 002672  002031                               BGE     5$                      ;Yes, error
5925 002674  005037  003154'                4$:     CLR     RESVAL                  ;Assume nothing
5926 002700  011000                               MOV     (R0),R0                 ;Nothing there, error
5927 002702  001425                               BEQ     5$                      ;Nothing there, finished
5928 002704  012737  000002  003154'               MOV     #2.,RESVAL              ;Assume array
5929 002712  005760  000002                       TST     2.(R0)                  ;Is it one?
5930 002716  002007                               BGE     11$                     ;Nope, either group or function
5931 002720  132760  000020  000003               BITB    #A.CON,A.BITS(R0)       ;Is it a label?
5932 002726  001413                               BEQ     5$                      ;Nope
5933 002730  005337  003154'                       DEC     RESVAL                  ;Yes, value is 1
5934 002734  000410                               BR      5$                      ;Go to it
```

```
 5935 002736  005237  003154'          11$:    INC     RESVAL                  ;Assume Function
 5936 002742  032760  040000  000002           BIT     #S.FLB,2.(R0)           ;Check the block
 5937 002750  001002                           BNE     5$                      ;It appears to be a function
 5938 002752  005237  003154'                  INC     RESVAL                  ;Otherwise, it must be an unknown
 5939 002756                           5$:     CALL    @RESCVT                 ;Store the result
 5940 002762  104402                           SYSERR                          ;Should never happen
 5941 002764  005366  000002                   DEC     2.(SP)                  ;Decrement loop count
 5942 002770  003310                           BGT     3$                      ;Loop until exhausted
 5943
 5944 002772  062706  000006           6$:     ADD     #6.,SP                  ;Pop garbage from the stack
 5945 002776                                   RETURN                          ;Finished
 5946
 5947 003000  104417                   7$:     DOMERR
 5948 003002  104411                   8$:     RANERR
 5950 003004                                   LIST
```

```
                                        MOPINS  BXNL,214
                                        .SBTTL  SYSTEM FUNCTION         NAME LIST

                                                .ENABLE LSB
003004  013737  003120' 003204'        MOV     MONBLK,RARGBL           ;Make argument into Right argument
003012  005037  005070'         1$:     CLR     OPTMP1                  ;Clear the search flag
003016  000421                          BR      4$                      ;And start the presses

003020  104417                  2$:     DOMERR                          ;Domain Error
003022  104411                  3$:     RANERR                          ;Rank Error
003024                                  LIST

003024                                  DOPINS  BXNL,214
                                        .SBTTL  SYSTEM FUNCTION         NAME LIST

                                        ;+
                                        ;  The temporary registers used are as follows
                                        ;
                                        ;       OPTMP1 - Flag indicating we have a list of initial characters to search
                                        ;       OPTMP2 - Name types to list, Bit 0-Labels, 1-Variables, 2-Functions
                                        ;       OPTMP3 - Maximum symbol length
                                        ;       OPTMP4 - Number of symbols to list
                                        ;       OPTMP9 - Action routine name for symbol table scanner routine
                                        ;-

003024  005737  003170'                 TST     LARGTY                  ;Is the Left argument charater type?
003030  001373                          BNE     2$                      ;No, Domain error
003032  013701  003162'                 MOV     LARGBL,R1               ;Point to the Left argument
003036  122761  000001  000002         CMPB    #1,A.DIMS(R1)           ;Is the Left argument Scalar or Vector?
003044  002766                          BLT     3$                      ;No, Rank error
003046  005761  000004                 TST     A.ALEN(R1)              ;Is Left argument null vector?
003052  001757                          BEQ     1$                      ;Yes, then handle like monadic function
003054  012737  000001  005070'        MOV     #1,OPTMP1               ;Set the search flag

003062  005037  005074'         4$:     CLR     OPTMP3                  ;Clear maximum symbol length
003066  005037  005076'                 CLR     OPTMP4                  ;Clear number of symbols in result
003072  013704  003204'                 MOV     RARGBL,R4               ;Point to the Right argument
003076  116402  000003                 MOVB    A.BITS(R4),R2           ;Get its bits
003102  042702  177771                 BIC     #^C<A.NUM>,R2           ;Is it numeric?
003106  001744                          BEQ     2$                      ;No, Domain error
003110  122764  000001  000002         CMPB    #1,A.DIMS(R4)           ;Is it a Scalar or Vector?
003116  002741                          BLT     3$                      ;No, Rank error
003120  016237  000000G 003220'        MOV     IRARGC(R2),RARGCV       ;Setup to convert Right arg to integer
003126  016237  000054' 003202'        MOV     LENTAB(R2),RARGFL       ;Setup Right argument length increment
003134  005037  005072'                 CLR     OPTMP2                  ;Clear the types flag
003140  016400  000004                 MOV     A.ALEN(R4),R0           ;Get number of elements
003144  001426                          BEQ     7$                      ;Right argument is null vector
003146  011403                          MOV     (R4),R3                 ;Point to Right argument data

003150                          5$:     CALL    @RARGCV                 ;Get a Right argument element
003154  013702  003222'                 MOV     RARGVA,R2               ;Is it less than one?
003160  003717                          BLE     2$                      ;Yes, Domain error
003162  022702  000003                 CMP     #3,R2                   ;Is it greater than three?
003166  002714                          BLT     2$                      ;Yes, Domain error
003170  005402                          NEG     R2                      ;Make it negative
003172  006302                          ASL     R2                      ;Times two for word offset
```

```
6010 003174  005004                        CLR     R4                      ;Clear a register
6011 003176  000261                        SEC                             ;And set the C-bit
6012 003200  000162  003212'               JMP     6$(R2)                  ;Convert type number to bit position
6013 003204  006104                        ROL     R4                      ;Function type
6014 003206  006104                        ROL     R4                      ;Variable type
6015 003210  006104                        ROL     R4                      ;Label type
6016 003212  050437  005072'        6$:    BIS     R4,OPTMP2               ;Save the types we want to list
6017 003216                                SOB     R0,5$                   ;Loop for all of Right argument
6018                                       .DSABL  LSB
6019
6020 003222  012737  003512' 005110' 7$:    MOV     #BXNLSH,OPTMP9          ;Set Scan action routine to search
6021 003230                                CALL    BXNLSN                  ;Pass one, scan table for eligible names
6022 003234  013701  005076'               MOV     OPTMP4,R1               ;Get the number of symbols in result
6023 003240                                MUL     OPTMP3,R1               ;Times the maximum length
6024 003254  103001                        BCC     8$                      ;Too big?
6025 003256  104403                        WSFULL                          ;Yes
6026 003260  012700  000012        8$:    MOV     #<A.DIM1+4>,R0          ;Get array block overhead
6027 003264  060100                        ADD     R1,R0                   ;Total number of bytes to get
6028 003266                                CALL    GETFRA                  ;Allocate space for result and fillin SI
6029 003272  010010                        MOV     R0,(R0)                 ;Fill in result array block
6030 003274  062720  000012               ADD     #<A.DIM1+4>,(R0)+       ;Pointer to data value
6031 003300  112720  000002               MOVB    #2,(R0)+                ;DIMS
6032 003304  112720  000200               MOVB    #A.ASB,(R0)+            ;BITS
6033 003310  010120                        MOV     R1,(R0)+                ;ALEN
6034 003312  013720  005076'               MOV     OPTMP4,(R0)+            ;DIM1 - Number of symbols
6035 003316  013720  005074'               MOV     OPTMP3,(R0)+            ;     - Maximum length of symbol
6036 003322  010037  003140'               MOV     R0,RESPTR               ;Pointer to result data
6037 003326  012737  003536' 005110'       MOV     #BXNLFL,OPTMP9          ;Set action routine to fill
6038 003334  005701                        TST     R1                      ;Any symbols in result?
6039 003336  001402                        BEQ     9$                      ;No, then skip filling in
6040 003340                                CALL    BXNLSN                  ;Pass two, fill in result
6041 003344                        9$:     RETURN                          ;All done
6042
6043 003346  012700  005344'       BXNLSN: MOV     #SYMB1,R0               ;Point to beginning of Symbol table
6044 003352  013701  003162'               MOV     LARGBL,R1               ;Point to the Left argument if any
6045
6046 003356  005002                1$:     CLR     R2                      ;Clear symbol type
6047 003360  016000  177776               MOV     S.AB(R0),R0             ;Get the next symbol
6048 003364  001001                        BNE     2$                      ;Last symbol?
6049 003366                                RETURN                          ;Yes, All done
6050 003370  011003                2$:     MOV     (R0),R3                 ;Get symbol's value
6051 003372  001771                        BEQ     1$                      ;No value, go get next symbol
6052 003374  132763  000100  000003       BITB    #F.FLB,A.BITS(R3)       ;Is it a function?
6053 003402  001403                        BEQ     3$                      ;No
6054 003404  052702  000004               BIS     #4,R2                   ;Mark the type as a function
6055 003410  000415                        BR      5$
6056 003412  132763  000200  000003 3$:    BITB    #A.ASB,A.BITS(R3)       ;Is it an array block?
6057 003420  001411                        BEQ     5$                      ;No (It must be a group)
6058 003422  132763  000020  000003       BITB    #A.CON,A.BITS(R3)       ;Is it a Label?
6059 003430  001403                        BEQ     4$                      ;No
6060 003432  052702  000001               BIS     #1,R2                   ;Mark the type as a Label
6061 003436  000402                        BR      5$
6062 003440  052702  000002        4$:     BIS     #2,R2                   ;Mark type as variable
6063 003444  030237  005072'        5$:     BIT     R2,OPTMP2               ;Do we want this type?
6064 003450  001742                        BEQ     1$                      ;No, then go get next symbol
6065 003452  005737  005070'               TST     OPTMP1                  ;Search flag set?
6066 003456  001003                        BNE     7$                      ;Yes, then must check initial char list
```

```
6067 003460                          6$:    CALL    @OPTMP9          ;Call action routine
6068 003464  000734                         BR      1$               ;Go get next symbol
6069
6070 003466  116002  000003          7$:    MOVB    S.CHR1(R0),R2    ;Get first character of the symbol
6071 003472  016104  000004                 MOV     A.ALEN(R1),R4    ;Get Left argument length
6072 003476  011103                         MOV     (R1),R3          ;Point to Left argument data
6073
6074 003500  122302                  8$:    CMPB    (R3)+,R2         ;Do the characters match?
6075 003502  001766                         BEQ     6$               ;Yes, then include the symbol
6076 003504                                 SOB     R4,8$            ;Loop for all of the Left argument
6077 003510  000722                         BR      1$               ;Not found, then go get new symbol
6078
6079 003512  126037  000002  005074' BXNLSH: CMPB    S.LEN(R0),OPTMP3 ;Which is larger, Symbol or max size?
6080 003520  003403                         BLE     1$               ;The maximum size so far
6081 003522  116037  000002  005074'        MOVB    S.LEN(R0),OPTMP3 ;The symbol, make it the new maximum
6082 003530  005237  005076'          1$:    INC     OPTMP4           ;Count this symbol
6083 003534                                 RETURN
6084
6085 003536  116004  000002          BXNLFL: MOVB    S.LEN(R0),R4     ;Get length of the symbol
6086 003542                                 PUSH    R4               ;Save it
6087 003544  010003                         MOV     R0,R3            ;Copy pointer to the symbol
6088 003546  062703  000003                 ADD     #S.CHR1,R3       ;Point to symbol characters
6089 003552  013702  003140'                MOV     RESPTR,R2        ;Point to result data area
6090
6091 003556  112322                  1$:    MOVB    (R3)+,(R2)+      ;Copy the symbol to the result
6092 003560                                 SOB     R4,1$            ;All of it
6093
6094 003564  013704  005074'                MOV     OPTMP3,R4        ;Get maximum length
6095 003570  162604                         SUB     (SP)+,R4         ;Compute amount of fill
6096 003572  001404                         BEQ     3$               ;No fill required
6097
6098 003574  112722  000040          2$:    MOVB    #CHRSPA,(R2)+    ;Blank fill
6099 003600                                 SOB     R4,2$            ;To the end
6100
6101 003604  010237  003140'          3$:    MOV     R2,RESPTR        ;Save pointer for next cycle
6102 003610                                 RETURN
6104 003612                                 LIST
```

```
   6106 003612                          DOPINS  FGET,150,T.DIM
   6330 003612                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 003612                          DOPINS  FPUT,152,T.DIM
   6485 003612                                  LIST
```

```
   6487 003612                          DOPINS  FPNT,154,T.DIM
   6714 003612                                  LIST
```

```
   6716 003612                          MOPINS  1SCN,130
   6723 003612                                  LIST
   6724
   6725
   6726 003612                          MOPINS  SCAN,62,T.DIM
   6732 003612                                  LIST
   6733
   6734
   6735 003612                          MOPINS  1RDT,124
   6741 003612                                  LIST
   6742
   6743
   6744 003612                          MOPINS  RDCT,60,T.DIM
   7044 003612                                  LIST
```

F 7

```
APLOO7  MACRO M1113  08-FEB-80 09:37  PAGE 85
SYSTEM FUNCTION          NAME LIST

   7046 003612                              MOPINS  GETIDN,0
   7193 003612                                      LIST
```

```
   7195 003612                          MOPINS  SATAN,0
   7404 003612                                  LIST
```

```
7406 003612                          MOPINS  OUTR,156
7476 003612                                  LIST
```

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 88
SYSTEM FUNCTION          NAME LIST

 7478 003612                            MOPINS  DOUTER,0
 7504 003612                                    LIST
 7505
 7506
 7507
 7508
 7509
 7510 003612                            MOPINS  PRDSET,0
 7564 003612                                    LIST
 7565
 7566
 7567
 7568
 7569
 7570 003612                            MOPINS  INSPAC,0
 7612 003612                                    LIST
```

```
   7614 003612                          DOPINS  INNR,156
```

7960 003612 LIST

APLOO7 MACRO M1113 08-FEB-80 09:37 PAGE 91
SYSTEM FUNCTION NAME LIST

7962 003612 MOPINS SUBR,0

APLOO7 MACRO M1113 08-FEB-80 09:37 PAGE 92

M 7

8725 003612 LIST

N 7

```
   8727 003612                          MOPINS  RAND,0
   8771 003612                                  LIST
```

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 94
SYSTEM FUNCTION          NAME LIST

      1          000001                              .END
```

SYMBOL TABLE

```
AC0    =%000000          CHRDIV= 000053          CS.BOO= 000010          D$BXRL  000574RG  020 D.UNON= 000006
AC1    =%000001          CHRDLR= 000176          CS.FLO= 000020          D.AND = 000000      D.$MAX= 000000
AC2    =%000002          CHRDLT= 000110          CS.IDN= 000040          D.BXCT= 000007      D.$MIN= 000000
AC3    =%000003          CHRDOT= 000056          CS.ONE= 000100          D.BXIO= 000007      D.$MUL= 000000
AC4    =%000004          CHREP = 000105          CTEPS   000152R 025     D.BXNL= 000007      D.$PLU= 000000
AC5    =%000005          CHRERR= 104431          CTMAX   000212R 025     D.BXPP= 000007      D.$RES= 000000
ALLINT  003274R 025      CHRFIL= 000177          CTRMAX  000172R 025     D.BXPW= 000007      D.$SUB= 000000
ARYRET  003276R 025      CHRFRO= 000135          C.ALOC= 000162          D.BXRL= 000007      D.1COM= 000001
ASCQUO= 000042           CHRLBR= 000073          C.BLOK= 000064          D.CIRC= 000001      D.1EXP= 000002
A.ALEN= 000004           CHRLPR= 000072          C.BLSZ= 000140          D.COMB= 000001      D.1ROT= 000003
A.ARG = 000040           CHRMNS= 000137          C.BUFF= 177760          D.COMP= 000001      EDICHR  005063R 025
A.ASB = 000200           CHRMUL= 000075          C.BUFL= 177762          D.CONT= 000006      EDICNT  005064R 025
A.BITS= 000003           CHRNEG= 000100          C.CCNT= 177766          D.CTLM= 000000      EDIFLG  005062R 025
A.CON = 000020           CHRPDL= 000271          C.CPTR= 177764          D.DEAL= 000003      EDIT  = 000003
A.DIMS= 000002           CHRPLU= 000055          C.CZ  = 001000          D.DECD= 000002      EIS   = 000000
A.DIM1= 000006           CHRPWR= 000120          C.DIG = 000040          D.DIV = 000000      ELESIZ  005132R 025
A.NUM = 000006           CHRQST= 000121          C.FASW= 020000          D.DMD = 000010      ENDCOR  000004R 025
A.PTR = 000000           CHRQUO= 000113          C.FLEN= 000010          D.DQAD= 177777      EOF   = 104475
A.TMP = 000010           CHRRBR= 000047          C.FLSH= 004000          D.DROP= 000002      ERASE = ****** GX
BADDIR= 104463           CHRRPR= 000042          C.FRC = 000001          D.ENCD= 000002      ERRMAX= 000166
BADERR= 104437           CHRSEM= 000074          C.IDEN= 000010          D.EQ  = 000000      ERRNUM  004772R 025
BLZBUB  005116R 025      CHRSMO= 000112          C.NAME= 000102          D.EXPD= 000002      ERRPTR  004770R 025
BRSELF  005046R 025      CHRSPA= 000040          C.NOBF= 000400          D.FGET= 000005      ERRO    004774R 025
BXCT    000000RG 002     CHRTAB= ****** GX       C.NUM = 000020          D.FMT = 000004      EXEBUF  005054R 025
BXIO    000010RG 002     CHRUPA= 000131          C.OP  = 000004          D.FPNT= 000005      ESCT    005056R 025
BXNLFL  003536R 020      CHR.BS= 000010          C.OPN = 010000          D.FPUT= 000005      ESPT    005060R 025
BXNLSH  003512R 020      CHR.CC= 000003          C.RDLN= 177776          D.GE  = 000000      FA.APD= 000100
BXNLSN  003346R 020      CHR.CR= 000015          C.REWT= 002000          D.GT  = 000000      FA.CRE= 000010
BXPP    000020RG 002     CHR.CZ= 000032          C.RND = 000010          D.IBM = 000002      FA.DLK= 001000
BXPW    000030RG 002     CHR.GT= 000046          C.SBIT= 177756          D.IND = 000002      FA.ENB= 100000
BXRL    000040RG 002     CHR.LF= 000012          C.SPFX= 000100          D.INNR= 000013      FA.EXC= 002000
CCNT    005072R 025      CHR.LT= 000043          C.SPGL= 000200          D.INTR= 000006      FA.EXT= 000004
CHANEX  003060R 025      CHR.SI= 000017          C.SPLX= 000001          D.LE  = 000000      FA.NSP= 000100
CHAN00  000256R 025      CHR.SL= 000057          C.SPRS= 000002          D.LOG = 000001      FA.NSR= 000200
CHAN01  000330R 025      CHR.SO= 000016          C.SRC = 177774          D.LT  = 000000      FA.POS= 010000
CHAN02  000512R 025      CHR.TB= 000011          C.SRCP= 000024          D.MAX = 000000      FA.RD = 000001
CHAN03  000674R 025      CHR.UL= 000106          C.TT  = 000004          D.MEM = 000002      FA.RWD= 004000
CHAN04  001056R 025      CHR..A= 000141          C.WIDC= 177770          D.MIN = 000000      FA.SEQ= 040000
CHAN05  001240R 025      CHR..E= 000145          C.WIDR= 177772          D.MUL = 000000      FA.SHR= 000040
CHAN06  001422R 025      CHR..S= 000163          DEBUG = 000000          D.NE  = 000000      FA.TMP= 000020
CHAN07  001604R 025      CHR..T= 000164          DEFERR= 104405          D.NN  = 000000      FA.WCK= 020000
CHAN08  001766R 025      CHR..U= 000165          DEVIL   005114R 025     D.NR  = 000000      FA.WRT= 000002
CHAN09  002150R 025      CHR..Z= 000172          DEVNTF= 104470          D.OR  = 000000      FD.BLK= 000010
CHAN10  002332R 025      CHR..0= 000060          DIGITS  000006R 025     D.PLUS= 000000      FD.CCL= 000002
CHAN11  002514R 025      CHR..1= 000061          DIMAX = 000020          D.PSUB= 000006      FD.COM= 020000
CHAN12  002676R 025      CH.AND= 000001          DIMBLK  003112R 025     D.PWR = 000001      FD.CR = 000002
CHAN13  000022R 023      CMDBUF  003250R 025     DIMPTR  003114R 025     D.QQAD= 000000      FD.DIR= 000010
CHAN14  000204R 023      CMDMAX= 000010          DIMTYP  003116R 025     D.QUAD= 000000      FD.FTN= 000001
CHNOPN= 104471           CNTSIZ  00513CR 025     DIRFUL= 104456          D.RES = 000000      FD.F11= 040000
CHRAND= 000051           COMBIN  005076R 025     DOMERR= 104417          D.RHO = 000000      FD.INS= 000010
CHRBEL= 000007           COMCNT  005126R 025     DPTERR= 104416          D.ROT = 000003      FD.ISP= 002000
CHRBKR= 000133           COMERR= 104425          DYDCNT  003224R 025     D.SCON= 000006      FD.MNT= 100000
CHRBOL= 000200           COMPTR  005122R 025     D$BXCT  000070RG 020    D.SDIF= 000006      FD.OSP= 004000
CHRBOX= 000114           COMRTN  003156R 025     D$BXIO  000322RG 020    D.SUB = 000000      FD.PLC= 000004
CHRCMA= 000054           COMSKP  005124R 025     D$BXNL  003024RG 020    D.SUBS= 000006      FD.PRN= 000004
CHRCOL= 000076           CORPTR  005120R 025     D$BXPP  000376RG 020    D.TAKE= 000002      FD.PSE= 010000
CHRDEL= 000107           CS.ASS= 000200          D$BXPW  000514RG 020    D.TRAN= 000004      FD.RAH= 000001
```

SYMBOL TABLE

```
FD.RAN= 000002          F.BKST= 000024          F.SPDV= 000072          LARGLE  003174R 025 M.BXEX= 000007
FD.REC= 000001          F.BKVB= 000064          F.SPUN= 000074          LARGLN  003166R 025 M.BXFX= 000007
FD.RWM= 000001          F.BR  = 000020          F.STBK= 000036          LARGPT  003164R 025 M.BXIO= 000007
FD.SDI= 000020          F.CHR = 000075          F.STK = 000022          LARGSF  003172R 025 M.BXLC= 000003
FD.SQD= 000040          F.CNTG= 000034          F.STOP= 000004          LARGTY  003170R 025 M.BXNC= 000007
FD.TTY= 000004          F.CODE= 000010          F.STP = 000001          LARGVA  003200R 025 M.BXNL= 000007
FD.WBH= 000002          F.DFNB= 000046          F.SUSP= 000002          LBINC   005100R 025 M.BXPP= 000007
FF.CHR= 000005          F.DSPT= 000044          F.TRC = 000002          LBUFF   003316R 025 M.BXPW= 000007
FF.NV = 000003          F.DVNM= 000134          F.UNIT= 000136          LBUFLN= 000621          M.BXRL= 000007
FF.POE= 000002          F.EFBK= 000010          F.URBD= 000020          LCOUNT  005106R 025 M.BXTS= 000007
FF.RWD= 000001          F.EFN = 000050          F.VAL = 177776          LENERR= 104412          M.BXWA= 000007
FF.RWF= 000006          F.END = 002000          F.VBN = 000064          LENTAB  000054R 025 M.CEI = 177777
FF.SPC= 000004          F.EOBB= 000032          F.VBSZ= 000060          LEXPTR  003100R 025 M.CHR = 000202
FHEADR  005012R 025     F.ERR = 000052          GETCOR= ****** GX       LFTOP   005066R 025 M.CMT = 000206
FHRET   005040R 025     F.FACC= 000043          GETCO1= ****** GX       LINC    005102R 025 M.DOUT= 000006
FILNTF= 104467          F.FFBY= 000014          GETFRA= ****** GX       LINE    005016R 025 M.DQAD= 000006
FIS   = 000000          F.FLAG= 000002          GETNB2= ****** GX       LININC  005026R 025 M.ENCD= 000004
FISCON= 046000          F.FLB = 000100          GETSCA= ****** GX       LISIZE= 000020          M.EXE = 000000
FIXPTR  003074R 025     F.FNAM= 000110          GNOME   003102R 025     LITOP   005072R 025 M.EXP = 000001
FLAGS   004764R 025     F.FNB = 000102          G.TICP= 000016          LNLNG = 104541          M.FACT= 000001
FLCBUF  003250R 025     F.FNDF= 000010          G.TICT= 000014          LOCLCT  005036R 025 M.FIL = 000177
FLINE   005014R 025     F.FTYP= 000116          G.TIDA= 000004          LOWCOR  000002R 025 M.FLR = 177777
FLNLST= ****** GX       F.FUN = 000010          G.TIHR= 000006          LSTERP  004766R 025 M.FMT = 000004
FLOEXP  003270R 025     F.FVER= 000120          G.TIMI= 000010          LWIDTH  000070R 025 M.GDUP= 000003
FLONEG  003272R 025     F.HEAD= 000006          G.TIMO= 000002          L.BITS= 000003          M.GDWN= 000003
FLSWTB  000222R 025     F.HIBK= 000004          G.TISC= 000012          L.CNT = 000002          M.GETI= 000006
FLTLEN= 000010          F.LBL = 000020          G.TIYR= 000000          L.PRNT= 000001          M.IBM = 000003
FNEND = ****** GX       F.LINE= 000012          HASHC   005136R 025     L.PTR = 000000          M.IDN = 000000
FNHEAD= ****** GX       F.LOC = 000004          HASHTB  005140R 025     L.PTR1= 000004          M.INSP= 000006
FNINSI  005044R 025     F.LSTC= 000004          HSHMSK= 177601          L.SUBS= 000002          M.IOTA= 000000
FNLINE= ****** GX       F.LUN = 000042          HSHSIZ= 000200          MACHIN= 000106          M.LIT = 000205
FORIGN  000032R 025     F.MBCT= 000054          ICHAN   005050R 025     MAXCHN= 000014          M.LN  = 000001
FO.APD= 000106          F.MBC1= 000055          IDIV  = ****** GX       MONBLK  003120R 025     M.MMD = 000010
FO.MFY= 000002          F.MBFG= 000056          ILFNAM= 104464          MONCVT  003132R 025     M.NEG = 000000
FO.RD = 000001          F.MON = 000002          ILLCHN= 104540          MONLEN  003130R 025     M.NOT = 000000
FO.UPD= 000006          F.NIL = 000001          ILLSW = 104565          MONPTR  003122R 025     M.NUB = 000006
FO.WRT= 000016          F.NRBD= 000024          IMUL  = ****** GX       MONSFT  003126R 025     M.NUM = 000204
FPMODE  000042R 025     F.NREC= 000030          INDERR= 104410          MONTYP  003124R 025     M.OUTR= 000012
FPP   = 000000          F.OPN = 000010          INNTMP  003250R 025     MONVAL  003134R 025     M.PITM= 000000
FPPLEN= 000004          F.OVBS= 000030          INNTM2  005016R 025     MSBXAV  000000RG 020    M.PRDS= 000006
FREBLK  007400R 025     F.PREV= 000000          INPTR   000064R 025     MSBXCR  001016RG 020    M.QQAD= 000000
FUZZ    000142R 025     F.PRNT= 100000          INSNAM= 104462          MSBXCT  000060RG 020    M.QUAD= 000000
FXPTR1  003076R 025     F.PROT= 010000          IOERR = 104477          MSBXEX  001600RG 020    M.RAND= 000012
FSEVAL= 000100          F.RACC= 000016          IRARGC= ****** GX       MSBXFX  002046RG 020    M.RAVL= 000000
FSEVI = 000200          F.RATT= 000001          I.CNT = 177776          MSBXIO  000312RG 020    M.RCIP= 000000
FSEXEC= 000400          F.RCNM= 000034          I.DIM = 177772          MSBXNC  002462RG 020    M.RDCT= 000011
FSFUNC= 001000          F.RCTL= 000017          I.LEN = 000012          MSBXNL  003004RG 020    M.REVS= 000003
F.ACTL= 000076          F.READ= 004000          I.OERR= 000062          MSBXPP  000366RG 020    M.ROLL= 000000
F.ALOC= 000040          F.REDF= 000040          I.OLD = 177770          MSBXPW  000504RG 020    M.SATA= 000001
F.BBFS= 000062          F.RES = 000040          I.PTR = 177774          MSBXRL  000564RG 020    M.SCAN= 000011
F.BDB = 000070          F.RET = 177772          I.STR = 177766          MSBXTS  000622RG 020    M.SHAP= 000000
F.BGBC= 000057          F.RSIZ= 000002          KILL  = ****** GX       MSBXWA  000732RG 020    M.SIGN= 000000
F.BITS= 177774          F.RTYP= 000000          LABERR= 104406          M.ABS = 000000          M.SUBR= 000001
F.BKDN= 000026          F.R2  = 000014          LABLPT  005042R 025     M.BOL = 000200          M.SYM = 000203
F.BKDS= 000020          F.R3  = 000016          LARGBL  003162R 025     M.BXAV= 000007          M.SYN = 000201
F.BKEF= 000050          F.R4  = 000020          LARGCV  003176R 025     M.BXCR= 000007          M.TRAN= 000012
F.BKP1= 000051          F.SEQN= 000100          LARGFL  003160R 025     M.BXCT= 000007          M.$ABS= 000000
```

APLOO7 MACRO M1113 08-FEB-80 09:37 PAGE 94-3
SYMBOL TABLE

```
M.$CEI= 000000          N.DID = 000024          RESCVT  003152R  025 SUBPTR  003110R  025 T.CON = 004000
M.$FLR= 000000          N.DVNM= 000032          RESERR= 104440       SYMBOL= ****** GX    T.DIM = 000100
M.$IDN= 000000          N.EXT = 000014          RESLEN  003150R  025 SYMB1   005344R  025 T.FUN = 010000
M.$NEG= 000000          N.FID = 000000          RESPTR  003140R  025 SYMF    005340R  025 T.LIST= 000400
M.$SIG= 000000          N.FNAM= 000006          RESSFT  003146R  025 SYMPT1  005346R  025 T.LIT = 000020
M.1RDT= 000011          N.FTYP= 000014          RESTYP  003144R  025 SYMTAB  005352R  025 T.MON = 000010
M.1REV= 000003          N.FVER= 000016          RESULT  003136R  025 SYMTBF= 104404       T.NIL = 000004
M.1SCN= 000011          N.NAME= 000006          RESVAL  003154R  025 SYMTOP  005350R  025 T.NUM = 002000
NAMBUF  003240R  025    N.NEXT= 000022          RINC    005104R  025 SYMTP   006272R  025 T.OPFN= 000001
NB.DEV= 000200          N.PPN = 000024          RLTBAS  005066R  025 SYMURN= 000054       T.OTHR= 040000
NB.DIR= 000100          N.STAT= 000020          ROTINC  005070R  025 SYNERR= 104407       T.SOP = 000002
NB.NAM= 000004          N.UNIT= 000034          ROUND   000010R  025 SYSER = ****** GX    T.SPOP= 000040
NB.SD1= 000400          OBUFLN= 001000          ROVER   003312R  025 SYSERR= 104402       T.TMP = 000200
NB.SD2= 001000          OCHAN   005052R  025 RSTS  = 000000       SYSTAT  000000R  024 T.VAL = 100000
NB.SNM= 000040          OMODE   000402R  023 RSX   = 000115       SY.BAD= 000001       T.VAR = 001000
NB.STP= 000020          OPTMP0  005066R  025 RTOP    005070R  025 SY.BRK= 100000       UPPER = 177400
NB.SVR= 000010          OPTMP1  005070R  025 RT11  = 000000       SY.BS = 000010       UTNAME  000344R  023
NB.TYP= 000002          OPTMP2  005072R  025 R.BITS= 000003       SY.CLR= 000040       VALERR= 104413
NB.VER= 000001          OPTMP3  005074R  025 R.FIX = 000001       SY.ERP= 000020       VARBLE  007400R  025
ND.DEV= 000200          OPTMP4  005076R  025 R.LCT = 000002       SY.NOU= 000100       VARBLS  000000R  025
ND.EXT= 000002          OPTMP5  005100R  025 R.LN  = 000002       SY.OPN= 000002       VERNUM  000000R  025
ND.NAM= 000004          OPTMP6  005102R  025 R.LNUM= 000010       SY.PMT= 001000       VRSION= 000002
ND.PPN= 000100          OPTMP7  005104R  025 R.NEXT= 000006       SY.SRT= 000400       WIDTH   000072R  025
ND.SEX= 000000          OPTMP8  005106R  025 R.OFF = 000000       SY.SVC= 000004       WROUND  000020R  025
ND.SES= 000020          OPTMP9  005110R  025 R.PREV= 000004       SY.SVL= 000200       WSDATE  000136R  025
ND.SNM= 000000          OPTM10  005112R  025 R.PTR = 000000       S.AB  = 177776       WSFULL= 104403
ND.SNS= 000040          OPTM14  005026R  025 R.SEQ = 000003       S.ASB = 100000       WSNAME  000074R  025
ND.SPN= 001400          ORIGIN  000030R  025 R.VAR = 000002       S.BFHD= 000020       WSTIME  000132R  025
NEGSGN  005063R  025    OUTPTR  000066R  025 R6    =%000006       S.BITS= 000002       $ADDF = ****** GX
NINPT1  003302R  025    OVROPR= 000007       R7    =%000007       S.CHR1= 000003       $CFCC = ****** GX
NINPT2  003300R  025    PATCH   007344R  025 SAME  = ****** GX    S.FATT= 000016       $CMPF = ****** GX
NINSTK  003304R  025    PRECSN  000052R  025 SCAMEM= ****** GX    S.FDB = 000140       $DIVF = ****** GX
NINTMP  003310R  025    RANDAC  003064R  025 SI    =%000005       S.FLB = 040000       $FIS  = ****** G
NINTYP  003306R  025    RANDY   003062R  025 SIDAMN= 104433       S.FNAM= 000006       $FPP  = ****** G
NOTAVL= 104472          RANERR= 104411       SIFRAM  003314R  025 S.FNB = 000036       $FPPLN= ****** G
NOTOPN= 104473          RARGBL  003204R  025 SISTAK  006350R  025 S.FNBW= 000017       $LDCIF= ****** GX
NPRT1   004762R  025    RARGCV  003220R  025 SITOP   006346R  025 S.FNTY= 000004       $LDF  = ****** GX
NULL    007364R  025    RARGFL  003202R  025 SITP    007144R  025 S.FTYP= 000002       $MACHN= ****** G
NUMCHN= 000020          RARGLE  003216R  025 SIWARN= 000200       S.LEN = 000002       $RSTS = ****** G
NUMDFL  003230R  025    RARGLN  003210R  025 SIZTAB  000044R  025 S.LINK= 177774       $RSX  = ****** G
NUMEFL  003233R  025    RARGPT  003206R  025 SI.BIT= 177774       S.NFEN= 000020       $RT11 = ****** G
NUMESN  003236R  025    RARGSF  003214R  025 SI.LEN= 177772       S.TYPE= 140000       $STF  = ****** GX
NUMFLG  003230R  025    RARGTY  003212R  025 SI.RET= 177772       S.VAL = 000000       $SUBF = ****** GX
NUMLEN  000062R  025    RARGVA  003222R  025 SI.VAL= 177776       TKDIMS  005072R  025 $$$OST= 000020
NUMNEG  003232R  025    RBASE   005074R  025 SKPSIZ  005134R  025 TKFRAM  005066R  025 .BOOLE= 000002
NUMQFL  003234R  025    RCOUNR  005112R  025 SM.BB = ****** GX    TKLEFT  005070R  025 .CHARA= 000000
NUMRIG  003231R  025    RCOUNT  005110R  025 SM.II = ****** GX    TMODE   000403R  023 .DEF. = 000002
NUMWID  003226R  025    REDCNT  005070R  025 SPHOLD  000004R  024 TMPTAG= 000166       .FLOAT= 000006
NUMZFL  003235R  025    REDIGN  005066R  025 SPINIT  000002R  024 TTYBUF  004140R  025 .FSRCB= ****** G
NYIERR= 104432          REDOFF  005072R  025 STKINC= 000774       TTYLEN= 000622       .INTEG= 000004
N.BLSZ= 000036          REENTR  003060R  025 STOP  = 104420       TURKEY= 104436       .MOLUN  000404R  023
N.DESC= 000020          REGTYP= 000001       SUBCNT  003104R  025 TYPE  = 000166       ...GBL= 000000
N.DEV = 000032          RESBIT  003142R  025 SUBDCT  003106R  025 T.BOL = 020000       ...TPC= 000020
N.DEVN= 000034

. ABS.  000000  000
        000000  001
```

F 8

```
APL007  MACRO M1113  08-FEB-80 09:37  PAGE 94-4
SYMBOL TABLE

TABLES  000050      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    003612      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13953 WORDS  ( 55 PAGES)
DYNAMIC MEMORY:  15534 WORDS  ( 59 PAGES)
ELAPSED TIME:  00:03:46
APL007,DM1:[305,301]APL007/-SP=APLRX1,APLDEF,APL007,APLOP,END
```

G 8

FILEID**APL008

```
  AAAAAA      PPPPPPPP      LL              000000        000000        888888
  AAAAAA      PPPPPPPP      LL              000000        000000        888888
AA      AA    PP      PP    LL            00      00    00      00    88      88
AA      AA    PP      PP    LL            00      00    00      00    88      88
AA      AA    PP      PP    LL            00      00    00    0000    88      88
AA      AA    PP      PP    LL            00      00    00    0000    88      88
AA      AA    PPPPPPPP      LL            00      00    00  00  00      888888
AA      AA    PPPPPPPP      LL            00      00    00  00  00      888888
AAAAAAAAAA    PP            LL            00      00    0000    00    88      88
AAAAAAAAAA    PP            LL            00      00    0000    00    88      88
AA      AA    PP            LL            00      00    00      00    88      88      ....
AA      AA    PP            LL            00      00    00      00    88      88      ....
AA      AA    PP            LLLLLLLLLL      000000        000000        888888        ....
AA      AA    PP            LLLLLLLLLL      000000        000000        888888        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 10:28
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    64-  3222    MIXED FUNCTION          MATRIX INVERT
```

APLOO8 MACRO M1113 08-FEB-80 10:28
TABLE OF CONTENTS

65- 3319 MIXED FUNCTION MATRIX DIVIDE (LEAST SQUARES FIT)

J 8


```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

K 8

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

DEFINITIONS ERRORS

```
                                .TITLE  APL008
                                .IDENT  \V02.03\

                                ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
                                ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
                                ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
                                ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
                                ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APL008 IS:
                                ;
                                ;       APL008 < ...,APL008,APLOP
                                ;
                                ;
000010                          OVROPR= 8.                      ;OVERLAY SPECIFIED OPERATORS
```

```
      5
      6                                 ;++
      7                                 ; FUNCTIONAL DESCRIPTION:
      8                                 ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
      9                                 ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
     10                                 ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
     11                                 ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
     12                                 ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
     13                                 ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
     14                                 ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
     15                                 ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
     16                                 ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
     17                                 ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
     18                                 ;       INTACT!
     19                                 ;--
     24
     27
    124
    125                                 ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
    126                                 ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
    127                                 ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
    128                                 ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
    129                                 ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
    130                                 ;MACROS.
    131
    132                                 .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
    133                                 .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
    134                                 .IF EQ OVROPR-D.'NAME
    135                                         .PSECT  CODE
    136                                 D$'NAME::
    137                                 .ENDM
    138
    139                                 .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
    140                                 .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
    141                                 .IF EQ OVROPR-M.'NAME
    142                                         .PSECT  CODE
    143                                         .IF NE NUM
    144                                 M$'NAME::
    145                                     .ENDC
    146                                 .ENDM
    147
    148                                 .MACRO  LIST
    149                                         .ENDC
    150                                 .ENDM
    151
    152                                 .MACRO  PRMRES
    153                                         .IF EQ 1
    154                                 .ENDM
    155
    159                                                 ;*********** ASSEMBLING THE FILE!!!!!
    160 007400                          PRMRES
    199 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 47
DEFINITIONS     ERRORS

    201 007400                          DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,,FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 47-1
DEFINITIONS     ERRORS

    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

DEFINITIONS ERRORS

```
641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.  P,.CHARACTER+CS.IDN
780 007400                                  LIST
781
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 49
DEFINITIONS     ERRORS

    783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    786 007400                                  LIST
    787
    788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    791 007400                                  LIST
    792
    793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
    799 007400                                  LIST
    800
    801
    802 007400                          MOPINS  $IDN,-2
    808 007400                                  LIST
    809
    810
    811 007400                          MOPINS  $FLR,-14
    855 007400                                  LIST
    856
    857
    858 007400                          MOPINS  $CEI,-16
    868 007400                                  LIST
    869
    870
    871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
    878 007400                                  LIST
    879
    880 007400                          MOPINS  $NEG,-4
    887 007400                                  LIST
    888
    889
    890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
    901 007400                                  LIST
    902
    903 007400                          MOPINS  $SIG,-6
    914 007400                                  LIST
    915
    916
    917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
    928 007400                                  LIST
    929
    930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
    939 007400                                  LIST
    940
    941
    942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
    949 007400                                  LIST
    950
    951
    952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
    959 007400                                  LIST
    960
    961 007400                          PRMRES
    966 007400                                  LIST
    967
    968
    969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
    977 007400                                  LIST
    978
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,.FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

DEFINITIONS ERRORS

```
1189 007400                          MOPINS  SHAP,102
1211 007400                                  LIST
1212
1213
1214
1215
1216
1217 007400                          DOPINS  RHO,102
1313 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

J 9

DEFINITIONS ERRORS

```
1598 007400                          MOPINS  IOTA,72
1632 007400                                  LIST
1633
1634
1635
1636
1637
1638 007400                          DOPINS  IND,72
1699 007400                                  LIST
```

K 9

K 9

DEFINITIONS ERRORS

```
1701 007400                          DOPINS  1COM,124
1709 007400                                  LIST
1710
1711
1712
1713
1714
1715 007400                          DOPINS  COMP,60,T.DIM
1789 007400                                  LIST
```

L 9

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

```
2151 007400                          MOPINS  1REV,146
2160 007400                                  LIST
2161
2162
2163
2164
2165
2166 007400                          MOPINS  REVS,144,T.DIM
2204 007400                                  LIST
```

DEFINITIONS ERRORS

```
2206 007400                          DOPINS  1ROT,146
2214 007400                                  LIST
2215
2216
2217
2218
2219
2220 007400                          DOPINS  ROT,144,T.DIM
2328 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 60
DEFINITIONS     ERRORS

   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 61
DEFINITIONS     ERRORS

   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

F 10

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 62
DEFINITIONS     ERRORS

   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

G 10

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 63
DEFINITIONS     ERRORS

   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```

```
                                        MOPINS  MMD,122
                                        .SBTTL  MIXED FUNCTION          MATRIX INVERT

                                        ;++
                                        ; The monadic form of domino finds the matrix inverse of its argument
                                        ; by calling the dyadic form with identity matrix with the same number
                                        ; of rows as the left argument.  A scalar argument is handled as a special
                                        ; case and returns the reciprocial of the argument. A zero argument causes
                                        ; a Domain error.
                                        ;--

000000  013704  003120'                         MOV     MONBLK,R4       ;Get pointer to the argument
000004  010437  003204'                         MOV     R4,RARGBL       ;Copy it to Right argument for D$DMD routine
000010  013737  003124' 003212'                 MOV     MONTYP,RARGTY   ;Copy the type too
000016  116402  000002                         MOVB    A.DIMS(R4),R2   ;Get the rank of the argument
000022  001541                                  BEQ     7$              ;Is it a scalar?
000024  020227  000002                         CMP     R2,#2           ;No, Is it greater than a matrix?
000030  003401                                  BLE     1$              ;No
000032  104411                                  RANERR                  ;Yes, Rank Error
000034  005764  000004           1$:          TST     A.ALEN(R4)      ;Is the argument a null array?
000040  001017                                  BNE     2$              ;No
000042                                          CALL    EMPTY           ;Yes, Return with equivalent null array
000046  126427  000002  000001                CMPB    A.DIMS(R4),#1   ;Is argument a null vector?
000054  001523                                  BEQ     6$              ;Yes, Then just return
000056  016502  177776                         MOV     SI.VAL(SI),R2   ;No, Get pointer to result array
000062  062702  000006                         ADD     #A.DIM1,R2      ;Point to dimension list of result array
000066  062704  000006                         ADD     #A.DIM1,R4      ;Point to dimension list of argument array
000072  012422                                  MOV     (R4)+,(R2)+     ;Copy the first dimension
000074  011412                                  MOV     (R4),(R2)       ;Copy the second dimension
000076  000512                                  BR      6$              ;Now we can return

000100  016401  000006           2$:          MOV     A.DIM1(R4),R1   ;Get the number of rows in argument
000104  010103                                  MOV     R1,R3           ;Save it for later
000106                                          MUL     R1,R1           ;Square it for size of identity matrix
000120  103001                                  BCC     3$              ;Too big?
000122  104403                                  WSFULL                  ;Yes, Workspace Full
000124  010100           3$:          MOV     R1,R0           ;Copy the identity matrix size
000126                                          ASH     PRECSN,R0       ;Times the element size
000142  010004                                  MOV     R0,R4           ;Save that too for later
000144  062700  000012                         ADD     #12,R0          ;Add in array block overhead
000150  162705  177772                         SUB     #SI.LEN,SI      ;Save the domino frame for D$DMD
000154                                          CALL    GETFRA          ;Go get space for the identity matrix
000160  010037  003162'                         MOV     R0,LARGBL       ;Copy pointer to Left argument for D$DMD routine
000164  012737  000006  003170'                 MOV     #.FLOAT,LARGTY  ;The type too
000172  010010                                  MOV     R0,(R0)         ;Fill in array block
000174  062720  000012                         ADD     #A.DIM1+4,(R0)+         ;Pointer to the data
000200  112720  000002                         MOVB    #2,(R0)+                ;Rank
000204  112720  000206                         MOVB    #<A.ASB+.FLOAT>,(R0)+   ;Bits
000210  010120                                  MOV     R1,(R0)+                ;ALEN
000212  010320                                  MOV     R3,(R0)+                ;First dimension
000214  010320                                  MOV     R3,(R0)+                ;Second dimension
000216  010001                                  MOV     R0,R1           ;Save pointer to the data
000220  105020           4$:          CLRB    (R0)+           ;Clear out identity matrix
000222                                          SOB     R4,4$           ;Real fast
000226                                          LDF     #^F1.0,AC0      ;Get a floating point one
000236  010304                                  MOV     R3,R4           ;Copy N, the number of rows
000240                                          ASH     PRECSN,R4       ;Times the element size
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 64-1
MIXED FUNCTION          MATRIX INVERT

 3278 000254  063704  000062'                ADD     NUMLEN,R4       ;Plus one = offset between diagonal elements
 3279 000260                          5$:     STF     AC0,(R1)        ;Fill in diagonal with ones
 3280 000266  060401                          ADD     R4,R1           ;Point to next diagonal element
 3281 000270                                  SOB     R3,5$           ;Loop for all the diagonal elements
 3282 000274                                  CALL    D$DMD           ;Use dyadic domino to compute inverse
 3283 000300                                  PUSH    <-(SI),-(SI),-(SI)>  ;Save the result SI frame
 3284 000306                                  CALL    KILL            ;Delete the identity matrix
 3285 000312  062705  177772                  ADD     #SI.LEN,SI      ;Delete the domino frame
 3286 000316                                  POP     <(SI)+,(SI)+,(SI)+>  ;Restore the result SI frame
 3287 000324                          6$:     RETURN
 3288
 3289                                 ;+
 3290                                 ; Scalar special case
 3291                                 ;-
 3292 000326  005737  003124'         7$:     TST     MONTYP          ;Is the argument numeric type?
 3293 000332  001001                          BNE     9$              ;Yes
 3294 000334  104417                  8$:     DOMERR                  ;Domain Error
 3295 000336  012700  000010          9$:     MOV     #A.DIM1+2,R0    ;Get array block overhead
 3296 000342  063700  000062'                ADD     NUMLEN,R0       ;Add size of floating point scalar
 3297 000346                                  CALL    GETFRA          ;Get space for the result
 3298 000352  152765  000006  177774        BISB    #.FLOAT,SI.BIT(SI)  ;Mark it as floating point
 3299 000360  010010                          MOV     R0,(R0)         ;Fill in array block
 3300 000362  062720  000010                  ADD     #A.DIM1+2,(R0)+         ;Pointer to data
 3301 000366  105020                          CLRB    (R0)+                   ;Rank
 3302 000370  112720  000206                  MOVB    #<A.ASB+.FLOAT>,(R0)+   ;Bits
 3303 000374  012720  000001                  MOV     #1,(R0)+                ;ALEN
 3304 000400  012720  000001                  MOV     #1,(R0)+                ;DIM1
 3305 000404  011402                          MOV     (R4),R2         ;Point to the argument data
 3306 000406  013701  003124'                MOV     MONTYP,R1       ;Get argument type
 3307 000412                                  CALL    @FMONCT(R1)     ;Fetch the argument
 3308 000416                                  TSTF    AC1             ;Is the argument zero?
 3309 000424                                  CFCC
 3310 000430  001741                          BEQ     8$              ;Yes, Then Domain error
 3311 000432                                  LDF     #^F1.0,AC0      ;Get a floating point one
 3312 000442                                  DIVF    AC1,AC0         ;Compute the reciprocial
 3313 000450                                  STF     AC0,(R0)        ;Store the result
 3314 000456                                  RETURN                  ;All done
 3316 000460                                  LIST
```

```
3318 000460                             DOPINS  DMD,122
3319                                    .SBTTL  MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)
3320
3321                                    ;++
3322                                    ; FUNCTIONAL DESCRIPTION:
3323                                    ;
3324                                    ; This function solves a linear system using Householder transformations
3325                                    ; to produce a least squares solution.  The algorithm is described in
3326                                    ; the following references:
3327                                    ;
3328                                    ;       M. A. Jenkins, 'The Solution of Linear Systems of Equations and
3329                                    ;               Linear Least Squares Problems in APL'', IBM Technical
3330                                    ;               Report No. 320-2989, June 1970
3331                                    ;
3332                                    ;       P. Businger & G. H. Golub, 'Linear Least Squares Solutions by
3333                                    ;               Householder Transformations'', Numerische Mathematik 7,
3334                                    ;               269-276, 1965
3335                                    ;
3336                                    ; The Right argument, A is m by n,
3337                                    ; the Left argument, B is m by p,
3338                                    ; and the Result, Z is n by p
3339                                    ;
3340                                    ; Written by Peter Haynes       Date: Feb, 1979
3341                                    ;
3342                                    ; Use of variables:
3343                                    ;
3344                                    ;       OPTMP0 - M, the number of rows
3345                                    ;       OPTMP1 - N, the number of columns in the Right argument, A
3346                                    ;       OPTMP2 - P, the number of columns in the Left argument, B
3347                                    ;       OPTMP3 - Pointer to the scaling vector, FACTOR
3348                                    ;       OPTMP4 - Pointer to the permutation vector, PP
3349                                    ;       OPTMP5 - Size of a row of A, = NxElement size
3350                                    ;       OPTMP6 - Column offset for A, = (MxN-1)xElement size
3351                                    ;       OPTMP7 - Size of a row of B and Z, = PxElement size
3352                                    ;       OPTMP8 - Temporary
3353                                    ;       OPTMP9 - Temporary
3354                                    ;       OPTM10 - Temporary, K
3355                                    ;       NUMLEN - Element size
3356                                    ;       RARGPT - Pointer to A
3357                                    ;       LARGPT - Pointer to B
3358                                    ;       RESPTR - Pointer to Z, the result
3359                                    ;
3360                                    ; Subroutines:
3361                                    ;
3362                                    ;       INNER - Calculates the a double precision inner product of two
3363                                    ;               vectors.
3364                                    ;
3365                                    ;       TRANS - Performs Householder transformations on matrix A or B.
3366                                    ;
3367                                    ;       SWAP  - Swaps a pair of rows or columns of a matrix.
3368                                    ;--
3369
3370                                    ;+
3371                                    ; Check for legal arguments
3372                                    ;-
3373 000460  013702  003204'                   MOV     RARGBL,R2       ;Point to the right argument
3374 000464  126227  000002  000002           CMPB    A.DIMS(R2),#2   ;Is the rank greater than a matrix?
```

```
3375 000472 003401                          BLE     2$              ;No, Ok then
3376 000474 104411                  1$:     RANERR                  ;Rank Error
3377 000476 013701 003162'          2$:     MOV     LARGBL,R1       ;Point to the left argument
3378 000502 126127 000002 000002            CMPB    A.DIMS(R1),#2   ;Is the rank greater than a matrix?
3379 000510 003371                          BGT     1$              ;Yes, Rank error then
3380 000512 026261 000006 000006            CMP     A.DIM1(R2),A.DIM1(R1)  ;Do the number of rows match?
3381 000520 001401                          BEQ     4$              ;Yes, Ok
3382 000522 104412                  3$:     LENERR                  ;Length Error
3383 000524 005737 003212'          4$:     TST     RARGTY          ;Is the right argument numeric?
3384 000530 001001                          BNE     6$              ;Yes, Ok
3385 000532 104417                  5$:     DOMERR                  ;Domain Error
3386 000534 005737 003170'          6$:     TST     LARGTY          ;Is the left argument numeric?
3387 000540 001774                          BEQ     5$              ;No, Domain error
3388 000542 005761 000004                   TST     A.ALEN(R1)      ;Is the left argument a null array?
3389 000546 001765                          BEQ     3$              ;Yes, Length error
3390 000550 016203 000004                   MOV     A.ALEN(R2),R3   ;Is the right argument a null array?
3391 000554 001762                          BEQ     3$              ;Yes, Length error
3392 000556 005303                          DEC     R3              ;Fudge the array length a little bit
3393 000560                                 ASH     PRECSN,R3       ;Times the element size
3394 000574 010337 005102'                  MOV     R3,OPTMP6       ;Equals the column offset for A
3395                                 ;+
3396                                 ; Have good arguments now, determine result shape and get space for it
3397                                 ;-
3398 000600 012703 000001                   MOV     #1,R3           ;Initialize for computing ALEN
3399 000604 010337 005070'                  MOV     R3,OPTMP1       ;Initialize for computing N
3400 000610 010337 005072'                  MOV     R3,OPTMP2       ;Initialize for computing P
3401 000614 005004                          CLR     R4              ;Initialize for computing result Rank
3402 000616 016237 000006 005066'           MOV     A.DIM1(R2),OPTMP0  ;Get M, the number of the rows
3403 000624 126227 000002 000001            CMPB    A.DIMS(R2),#1   ;What is the shape of right argument
3404 000632 003410                          BLE     7$              ;A scalar or vector
3405 000634 016203 000010                   MOV     A.DIM1+2(R2),R3 ;A matrix, get N
3406 000640 010337 005070'                  MOV     R3,OPTMP1       ;Save N, the number of columns in right arg
3407 000644 005204                          INC     R4              ;The result Rank is at least a vector
3408 000646 023703 005066'                  CMP     OPTMP0,R3       ;Is M<N, i.e. columns linearly dependent?
3409 000652 002723                          BLT     3$              ;Yes, Length error
3410 000654 126127 000002 000001    7$:     CMPB    A.DIMS(R1),#1   ;What is the shape of the left argument?
3411 000662 002415                          BLT     8$              ;Scalar
3412 000664 001416                          BEQ     9$              ;Vector
3413 000666 016137 000010 005072'           MOV     A.DIM1+2(R1),OPTMP2  ;Matrix, Save P, number of columns of B
3414 000674                                 MUL     OPTMP2,R3       ;The number of elements in result, ALEN
3415 000710 103525                          BCS     13$             ;Workspace full
3416 000712 005204                          INC     R4              ;Rank is one greater than before
3417 000714 000402                          BR      9$
3418 000716 105704                  8$:     TSTB    R4              ;Is the left argument a scalar?
3419 000720 001300                          BNE     3$              ;No, Length error
3420 000722 010300                  9$:     MOV     R3,R0           ;Copy the size of the result
3421 000724                                 ASH     PRECSN,R0       ;Multiply by the element size
3422 000740                                 PUSH    R4              ;Copy the result Rank
3423 000742                                 CONINC  R4              ;Cheat for scalars
3424 000746 006304                          ASL     R4              ;Make it in words
3425 000750 062704 000006                   ADD     #A.DIM1,R4      ;Add in overhead
3426 000754 060400                          ADD     R4,R0           ;Total space to get
3427 000756                                 CALL    GETFRA          ;Get space for result and build SI frame
3428 000762 152765 000006 177774            BISB    #.FLOAT,SI.BIT(SI)  ;Mark it as floating point
3429 000770 060420                          ADD     R4,(R0)+        ;Fill in array block, Pointer to data
3430 000772                                 POP     R4              ;Retrieve result Rank
3431 000774 110420                          MOVB    R4,(R0)+                ;DIMS
```

```
3432 000776 112720 000206                  MOVB    #<A.ASB+.FLOAT>,(R0)+   ;BITS
3433 001002 010320                         MOV     R3,(R0)+                ;ALEN
3434 001004 013720 005072'                 MOV     OPTMP2,(R0)+            ;DIM1, Assume P
3435 001010 023727 005070' 000001          CMP     OPTMP1,#1               ;Is N = 1?
3436 001016 001403                         BEQ     10$                     ;Yes, We guessed right
3437 001020 013760 005070' 177776          MOV     OPTMP1,-2(R0)           ;No, So use N
3438 001026 020427 000001          10$:    CMP     R4,#1           ;What is the result Rank?
3439 001032 003035                         BGT     11$             ;A matrix
3440 001034 001436                         BEQ     12$             ;A vector
3441 001036 105762 000002                  TSTB    A.DIMS(R2)      ;A scalar, Are both arguments scalars?
3442 001042 001033                         BNE     12$             ;No
3443 001044 013704 003170'                 MOV     LARGTY,R4       ;Yes, Special case, Get left argument type
3444 001050 013701 003212'                 MOV     RARGTY,R1       ;Get right argument type
3445 001054 013702 003164'                 MOV     LARGPT,R2       ;Point to left argument data
3446 001060 013703 003206'                 MOV     RARGPT,R3       ;Point to right argument type
3447 001064                                CALL    @FLARGC(R4)     ;Fetch the left argument
3448 001070                                CALL    @FRARGC(R1)     ;Fetch the right argument
3449 001074                                TSTF    AC2             ;Is right argument zero?
3450 001102                                CFCC                    ; i.e. the divisor
3451 001106 001611                         BEQ     5$              ;Yes, Domain error
3452 001110                                DIVF    AC2,AC1         ;This is domino for scalars!
3453 001116                                STF     AC1,(R0)        ;Store the result
3454 001124                                RETURN                  ;All done, that was real easy
3455
3456 001126 013720 005072'         11$:    MOV     OPTMP2,(R0)+    ;Store the second dimension, P
3457 001132 010037 003140'         12$:    MOV     R0,RESPTR       ;Save the pointer to the result data
3458                               ;+
3459                               ; Now get space for the temporary vectors: FACTOR, floating point, and
3460                               ; PP, pointers, both of length N.
3461                               ;-
3462 001136 013701 000062'                 MOV     NUMLEN,R1       ;Get element size for Factor
3463 001142 062701 000002                  ADD     #2,R1           ;Two more bytes for permutation vector
3464 001146                                MUL     OPTMP1,R1       ;And there are N elements
3465 001162 103001                         BCC     14$             ;Not too big?
3466 001164 104403                 13$:    WSFULL                  ;Workspace full
3467 001166 012700 000004          14$:    MOV     #4,R0           ;Array block overhead
3468 001172 060100                         ADD     R1,R0           ;Total amount to get
3469 001174                                CALL    GETTRA          ;Get the temporery space
3470 001200 005720                         TST     (R0)+           ;Skip over the pointer field
3471 001202 012720 100000                  MOV     #S.ASB,(R0)+    ;Mark it as an array block
3472 001206 010037 005074'                 MOV     R0,OPTMP3       ;Save pointer to Factor
3473 001212 163701 005070'                 SUB     OPTMP1,R1       ;Get the size of Factor
3474 001216 163701 005070'                 SUB     OPTMP1,R1       ; in words
3475 001222 010137 005100'                 MOV     R1,OPTMP5       ;Save it as size of a row of A
3476 001226 060100                         ADD     R1,R0           ;Point to the permutation vector
3477 001230 010037 005076'                 MOV     R0,OPTMP4       ;Save the permutation vector pointer
3478                               ;+
3479                               ; Now we're ready to initialize FACTOR and PP(permutation pointers)
3480                               ;-
3481 001234 013704 003140'                 MOV     RESPTR,R4       ;Point to the result
3482 001240 013702 005074'                 MOV     OPTMP3,R2       ;Point to Factor
3483 001244 013703 005070'                 MOV     OPTMP1,R3       ;Set count to N
3484 001250 013701 000062'                 MOV     NUMLEN,R1       ;Get element size
3485 001254                                MUL     OPTMP2,R1       ;Times P = size of result row
3486 001270 010137 005104'                 MOV     R1,OPTMP7       ;Save the result row size
3487 001274                        15$:    CLRF    (R2)+           ;Clear Factor
3488 001302 010420                         MOV     R4,(R0)+        ;Initialize PP
```

```
3489 001304  060104                        ADD     R1,R4           ;Point to the next result row
3490 001306                                SOB     R3,15$          ;Loop for N elements
3491                                ;+
3492                                ; Copy the arguments if necessary since they will be destoryed
3493                                ;-
3494 001312  012701  000006                MOV     #.FLOAT,R1      ;Convert the arguments to floating point
3495 001316  010502                        MOV     SI,R2           ;Point to current SI frame
3496 001320  062702  177764                ADD     #<2*SI.LEN>,R2  ;Back up two parse frames to Left arg frame
3497 001324                                CALL    CONVERT         ;Copy and convert it to floating point
3498 001330  011037  003164'               MOV     (R0),LARGPT     ;Save the pointer to the new left argument
3499 001334  062702  177764                ADD     #<2*SI.LEN>,R2  ;Point to the Right argument parse frame
3500 001340                                CALL    CONVERT         ;Copy and convert it to floating point
3501 001344  011000                        MOV     (R0),R0         ;Get pointer to new right argument data
3502 001346  010037  003206'               MOV     R0,RARGPT       ;Save the pointer to the new right argument
3503                                ;+
3504                                ; Find the largest value in each column of A and compute FACTOR
3505                                ;-
3506 001352  013704  005066'               MOV     OPTMP0,R4       ;Get M, the number of rows in A
3507 001356  013701  005100'               MOV     OPTMP5,R1       ;Get the size of a row of A
3508 001362  013702  005070'       16$:    MOV     OPTMP1,R2       ;Get N, the number of columns to scan
3509 001366                                CLRF    AC0             ;Clear the current max
3510 001374                        17$:    LDF     (R0)+,AC1       ;Get an element of A
3511 001402                                ABSF    AC1             ;Get its absolute value
3512 001410                                CMPF    AC1,AC0         ;Is it larger than the current max?
3513 001416                                CFCC
3514 001422  003403                        BLE     18$             ;No
3515 001424                                STF     AC1,AC0         ;Yes, Use it as new max
3516 001432                        18$:    SOB     R2,17$          ;Loop for each element in the row
3517 001436  160100                        SUB     R1,R0           ;Back up to the beginning of the row
3518 001440  013702  005070'               MOV     OPTMP1,R2       ;Get N, the number of columns to scan
3519 001444  013703  005074'               MOV     OPTMP3,R3       ;Get pointer to FACTOR
3520 001450                        19$:    LDF     (R0)+,AC1       ;Get an element of A
3521 001456                                DIVF    AC0,AC1         ;Compute  A(i,j)/max
3522 001464                                ABSF    AC1             ;Get its absolute value
3523 001472                                CMPF    (R3)+,AC1       ;Is FACTOR less than it?
3524 001500                                CFCC
3525 001504  002005                        BGE     20$             ;No
3526 001506  163703  000062'               SUB     NUMLEN,R3       ;Back up to FACTOR(j)
3527 001512                                STF     AC1,(R3)+       ;Yes, Use it for FACTOR(j)
3528 001520                        20$:    SOB     R2,19$          ;Loop for each element in the row
3529 001524                                SOB     R4,16$          ;Loop for each row
3530                                ;+
3531                                ; Scale A
3532                                ;-
3533 001530  013702  005070'               MOV     OPTMP1,R2       ;Get N, the number of columns
3534 001534  013700  003206'               MOV     RARGPT,R0       ;Point to A
3535 001540  013703  005074'               MOV     OPTMP3,R3       ;Point to FACTOR
3536 001544                        21$:    LDF     (R3)+,AC0       ;Get an element of FACTOR
3537 001552  013704  005066'               MOV     OPTMP0,R4       ;Get M, the number of rows
3538 001556                        22$:    LDF     (R0),AC1        ;Get an element of A
3539 001564                                DIVF    AC0,AC1         ;Scale A
3540 001572                                STF     AC1,(R0)        ;Store A
3541 001600  060100                        ADD     R1,R0           ;Point to next A element, in next row
3542 001602                                SOB     R4,22$          ;Loop for each element in the column
3543 001606  163700  005102'               SUB     OPTMP6,R0       ;Back up to next column
3544 001612                                SOB     R2,21$          ;Loop for each column
3545                                ;+
```

```
3546                                    ; Start the main Householder transformation loop
3547                                    ;-
3548 001616                                     PUSH    LARGPT          ;Save pointer to B(k,1)
3549 001622                                     PUSH    RARGPT          ;Save pointer to A(k,k)
3550 001626  005037  005112'                    CLR     OPTM10          ;K = 0
3551 001632  013701  005112'            HOUSE:  MOV     OPTM10,R1       ;Get K
3552 001636  010137  005106'                    MOV     R1,OPTMP8       ;Set TI = K
3553 001642  010137  005110'                    MOV     R1,OPTMP9       ;Set TJ = K
3554 001646  010100                             MOV     R1,R0           ;Set L = K
3555 001650  010103                             MOV     R1,R3           ;Set I = K
3556 001652                                     ASH     PRECSN,R1       ;Convert to column offset
3557 001666  011602                             MOV     (SP),R2         ;Get pointer to A(k,k)
3558 001670                                     CLRF    AC2             ;MV = 0.0
3559                                    ;+
3560                                    ; Look for the maximum remaining element to use as a pivot
3561                                    ;-
3562 001676                            23$:     CLRF    AC0             ;T = 0.0
3563 001704  013704  005112'                    MOV     OPTM10,R4       ;J = K
3564 001710                            24$:     LDF     (R2)+,AC1       ;Get A(i,j)
3565 001716                                     ABSF    AC1             ;Get its absolute value
3566 001724                                     CMPF    AC0,AC1         ;Is T .ge .ab A(i,j) ?
3567 001732                                     CFCC
3568 001736  002004                             BGE     25$             ;Yes
3569 001740                                     STF     AC1,AC0         ;No, Use A(i,j) as new T
3570 001746  010400                             MOV     R4,R0           ;L = J
3571 001750  005204                    25$:     INC     R4              ;Next J
3572 001752  020437  005070'                    CMP     R4,OPTMP1       ;End of a row?
3573 001756  002754                             BLT     24$             ;No
3574
3575 001760                                     CMPF    AC2,AC0         ;MV .ge T ?
3576 001766                                     CFCC
3577 001772  002007                             BGE     26$             ;Yes
3578 001774                                     STF     AC0,AC2         ;No, MV = T
3579 002002  010337  005106'                    MOV     R3,OPTMP8       ;TI = I
3580 002006  010037  005110'                    MOV     R0,OPTMP9       ;TJ = L
3581 002012  060102                    26$:     ADD     R1,R2           ;Next row
3582 002014  005203                             INC     R3              ;Next I
3583 002016  020337  005066'                    CMP     R3,OPTMP0       ;End of the columns?
3584 002022  002725                             BLT     23$             ;No
3585                                    ;+
3586                                    ; Is the pivot canidate large enough?
3587                                    ;-
3588 002024                                     CMPF    FUZZ,AC2        ;MV<Fuzz, i.e. Is the array singular?
3589 002034                                     CFCC
3590 002040  002401                             BLT     27$             ;No
3591 002042  104417                             DOMERR                  ;Yes, Domain Error
3592                                    ;+
3593                                    ; Swap the TJ and K-th columns. the TI and K-th rows of A, and the TJ and K-th
3594                                    ; elements of PP.
3595                                    ;-
3596 002044                            27$:     PUSH    RARGPT          ;Get pointer to A
3597 002050  023737  005112' 005110'            CMP     OPTM10,OPTMP9   ;Is K = TJ ?
3598 002056  001427                             BEQ     29$             ;Yes, Then don't need to swap columns or PP
3599
3600 002060  013700  005066'                    MOV     OPTMP0,R0       ;Get M
3601 002064  013701  005110'                    MOV     OPTMP9,R1       ;Get TJ
3602 002070  013702  000062'                    MOV     NUMLEN,R2       ;Get column offset
```

```
3603 002074  013704  005100'                  MOV     OPTMP5,R4       ;Get the length of a row
3604 002100                                   CALL    SWAP            ;Swap K-th and TJ-th columns of A
3605
3606 002104  013701  005110'                  MOV     OPTMP9,R1       ;Get TJ
3607 002110  006301                           ASL     R1              ;Convert to offset
3608 002112  063701  005076'                  ADD     OPTMP4,R1       ;Convert to pointer to PP(tj)
3609 002116  013703  005112'                  MOV     OPTM10,R3       ;Get K
3610 002122  006303                           ASL     R3              ;Convert to offset
3611 002124  063703  005076'                  ADD     OPTMP4,R3       ;Convert to pointer to PP(k)
3612 002130  011100                           MOV     (R1),R0         ;Now swap the elements of PP
3613 002132  011311                           MOV     (R3),(R1)       ; PP(tj) and
3614 002134  010013                           MOV     R0,(R3)         ; PP(k)
3615
3616 002136  023737  005112' 005106' 29$:     CMP     OPTM10,OPTMP8   ;Is K = TI ?
3617 002144  001424                           BEQ     32$             ;Yes, Then skip interchange of rows
3618
3619 002146  013700  005070'                  MOV     OPTMP1,R0       ;Get N
3620 002152  013701  005106'                  MOV     OPTMP8,R1       ;Get TI
3621 002156  013702  005100'                  MOV     OPTMP5,R2       ;Get row length
3622 002162  013704  000062'                  MOV     NUMLEN,R4       ;Get column offset
3623 002166                                   CALL    SWAP            ;Swap the K-th and TI-th rows of A
3624
3625 002172  013716  003164'                  MOV     LARGPT,(SP)     ;Get B
3626 002176  013700  005072'                  MOV     OPTMP2,R0       ;Get P
3627 002202  013701  005106'                  MOV     OPTMP8,R1       ;Get TI
3628 002206  013702  005104'                  MOV     OPTMP7,R2       ;Get row length
3629 002212                                   CALL    SWAP            ;Swap the K-th and TI-th rows of B
3630
3631 002216  005726                  32$:     TST     (SP)+           ;Clean matrix pointer off stack
```

```
3633                                    ;+
3634                                    ; Now we can get to the meat of the transformations
3635                                    ;-
3636 002220  013703  005100'                MOV     OPTMP5,R3       ;Get the increment for vector 1
3637 002224  010304                         MOV     R3,R4           ;Get the increment for vector 2
3638 002226  013700  005066'                MOV     OPTMP0,R0       ;Get M
3639 002232  163700  005112'                SUB     OPTM10,R0       ;The number of elements in the product
3640 002236  011601                         MOV     (SP),R1         ;Get pointer to K-th column of A
3641 002240  010102                         MOV     R1,R2           ;Copy pointer to K-th column of A
3642 002242                                 CALL    INNER           ;Get the inner product of K-th column of A
3643 002246                                 STF     AC0,AC4         ;Copy it to T
3644 002254                                 CALL    .SQRT           ;Get ALFA = sqrt(T)
3645 002260                                 LDF     @(SP),AC1       ;Get MV = A(k,k)
3646 002270                                 CFCC                    ;Is A(k,k) positive?
3647 002274  002403                         BLT     33$             ;No
3648 002276                                 NEGF    AC0             ;Yes, Then ALFA = -ALFA
3649 002304                          33$:   STF     AC0,AC3         ;Save it for later
3650 002312                                 STF     AC1,AC2         ;Copy A(k,k)
3651 002320                                 SUBF    AC0,AC2         ;A(k,k) - ALFA
3652 002326                                 STF     AC2,@(SP)       ;Save it in A(k,k)
3653 002336                                 MULF    AC0,AC1         ;MV * A(k,k)
3654 002344                                 LDF     AC4,AC2         ;Get T into a FAC we can use
3655 002352                                 SUBF    AC1,AC2         ;T = T - MV * A(k,k)
3656 002360                                 STF     AC2,AC4         ;Now put T back where it was
3657
3658 002366  011637  005110'                MOV     (SP),OPTMP9     ;Get pointer to A(k,k)
3659 002372  063737  000062' 005110'        ADD     NUMLEN,OPTMP9   ;Point to K+1st column
3660 002400  013737  005112' 005106'        MOV     OPTM10,OPTMP8   ;J = K
3661 002406                                 PUSH    OPTMP1          ;Get number of columns to process
3662 002412                                 CALL    TRANS           ;Apply Householder transforms to A
3663
3664 002416  016637  000004  005110'        MOV     4(SP),OPTMP9    ;Get pointer to first column of B
3665 002424  013737  177777  005106'        MOV     #-1,OPTMP8      ;J = 1
3666 002432  013716  005072'                MOV     OPTMP2,(SP)     ;Number of columns to process
3667 002436  013704  005104'                MOV     OPTMP7,R4       ;Get inner product column increment for B
3668 002442                                 CALL    TRANS           ;Apply Householder transforms to B
3669 002446  005726                         TST     (SP)+           ;Clean up stack
3670
3671 002450                                 STF     AC3,@(SP)       ;A(k,k) = ALFA
3672 002460  063766  005104' 000002        ADD     OPTMP7,2(SP)    ;Next row of B
3673 002466  063716  005100'                ADD     OPTMP5,(SP)     ;Next A(k,k)
3674 002472  063716  000062'                ADD     NUMLEN,(SP)     ;All the way
3675 002476  005237  005112'                INC     OPTM10          ;Next K
3676 002502  023737  005112' 005070'        CMP     OPTM10,OPTMP1   ;Done yet?
3677 002510  002002                         BGE     39$             ;Yes
3678 002512  000137  001632'                JMP     HOUSE           ;No
3679 002516  022626                   39$:  CMP     (SP)+,(SP)+     ;Clean up the stack
3680                                    ;+
3681                                    ; Now do the back substitutions
3682                                    ;-
3683 002520  013700  003140'          BACK: MOV     RESPTR,R0       ;Point to the result
3684 002524  013701  005070'                MOV     OPTMP1,R1       ;Get N
3685 002530                                 MUL     OPTMP2,R1       ;Times P = number of elements
3686 002544                                 ASH     PRECSN,R1       ;Times element size = total size in bytes
3687 002560  105020                   1$:   CLRB    (R0)+           ;Zero the result
3688 002562                                 SOB     R1,1$           ;Real fast
3689 002566  013701  005070'                MOV     OPTMP1,R1       ;Get N
```

```
APLOO8  MACRO M1113  08-FEB-80 10:28  PAGE 66-1
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

 3690 002572  005301                           DEC     R3              ;Fudge N a little bit
 3691 002574  010137  005112'                  MOV     R1,OPTM10       ;I = N
 3692 002600  010103                           MOV     R1,R3           ;Copy it
 3693 002602                                   MUL     OPTMP5,R1       ;Offset to row I of A
 3694 002616  063701  003206'                  ADD     RARGPT,R1       ;Convert to pointer to row I of A
 3695 002622  010137  005106'                  MOV     R1,OPTMP8       ;Save it
 3696 002626  063737  005100' 005106'         ADD     OPTMP5,OPTMP8   ;Point to the next row
 3697 002634  063737  005100' 005106'         ADD     OPTMP5,OPTMP8   ;Point to the end of the row, A(i,i)
 3698 002642                                   MUL     OPTMP7,R3       ;Offset to row I of B
 3699 002656  063703  003164'                  ADD     LARGPT,R3       ;Convert to pointer to row I of B
 3700
 3701 002662  013702  005112'          2$:    MOV     OPTM10,R2       ;Get I
 3702 002666  006302                           ASL     R2              ;Convert to word offset to PP(i)
 3703 002670  063702  005076'                  ADD     OPTMP4,R2       ;Convert to pointer to PP(i)
 3704 002674  011202                           MOV     (R2),R2         ;Get pointer to row PP(i) of Z
 3705 002676  013704  005106'                  MOV     OPTMP8,R4       ;Get pointer to previous A(i,i)
 3706 002702  163704  005100'                  SUB     OPTMP5,R4       ;Back up one row
 3707 002706  163704  000062'                  SUB     NUMLEN,R4       ;Back up one element
 3708 002712  010437  005106'                  MOV     R4,OPTMP8       ;Save pointer to new A(i,i)
 3709 002716                                   LDF     (R4),AC3        ;Get A(i,i)
 3710 002724  005004                           CLR     R4              ;Column offset for Z
 3711 002726  013700  005072'                  MOV     OPTMP2,R0       ;J = P
 3712                                  ;+
 3713                                  ; Do special inner product, rows of A with columns of Z
 3714                                  ;-
 3715 002732                           3$:     SETD                    ;Force double precision mode
 3716 002736                                   PUSH    <R0,R1,R2,R3>   ;Save some registers
 3717 002746  013700  005070'                  MOV     OPTMP1,R0       ;Get the number of elements in product
 3718 002752  013702  005076'                  MOV     OPTMP4,R2       ;Get pointer to PP vector
 3719 002756                                   CLRF    AC0             ;For result of inner product
 3720 002764  012203                   4$:     MOV     (R2)+,R3        ;Point to Z(PP(i)), the next row of Z
 3721 002766  060403                           ADD     R4,R3           ;Add column offset so pointing to Z(PP(i),j)
 3722 002770  023727  000062' 000004          CMP     NUMLEN,#4       ;Are we in single precision mode?
 3723 002776  001007                           BNE     5$              ;No, good
 3724 003000                                   LDCFD   (R1),AC1        ;Convert A(i,j) to double precision
 3725 003006                                   LDCFD   (R3),AC2        ;Convert Z(PP(i),j) to double precision
 3726 003014  000406                           BR      6$
 3727 003016                           5$:     LDF     (R1),AC1        ;Get A(i,j)
 3728 003024                                   LDF     (R3),AC2        ;Get Z(PP(i),j)
 3729 003032                           6$:     MULF    AC2,AC1         ;Form the product
 3730 00304C                                   ADDF    AC1,AC0         ;Accumulate the sum
 3731 003046  063701  000062'                  ADD     NUMLEN,R1       ;Next column of A
 3732 003052                                   SOB     R0,4$           ;Loop for all the elements
 3733 003056  023727  000062' 000004          CMP     NUMLEN,#4       ;Are we in single precision mode?
 3734 003064  001005                           BNE     7$              ;No, Good we can return
 3735 003066                                   SETF                    ;Yes, Go back to single precision mode
 3736 003072                                   LDCDF   AC0,AC0         ;COnvert result to single precision
 3737 003100                           7$:     POP     <R3,R2,R1,R0>   ;Restore the registers
 3738
 3739 003110                                   LDF     (R3)+,AC1       ;Get B(i,j)
 3740 003116                                   SUBF    AC0,AC1         ;B(i,j) - A(i,j).Z(PP(i),j)
 3741 003124                                   DIVF    AC3,AC1         ;(B(i,j) - A(i,j).Z(PP(i),j)) / A(i,i)
 3742 003132                                   STF     AC1,(R2)+       ;Save it in Z(PP(i),j)
 3743 003140  063704  000062'                  ADD     NUMLEN,R4       ;Next column of Z
 3744 003144  005300                           DEC     R0              ;Loop for each column of Z (This should
 3745 003146  003402                           BLE     8$              ; be SOB R0,3$ but 3$ is too far away)
 3746 003150  000137  002732'                  JMP     3$
```

```
3747
3748 003154  163701  005100'          8$:     SUB     OPTMP5,R1       ;Back up one row of A
3749 003160  163703  005104'                  SUB     OPTMP7,R3       ;Back up one row of B
3750 003164  163703  005104'                  SUB     OPTMP7,R3
3751 003170  005337  005112'                  DEC     OPTM10          ;I = I - 1
3752 003174  002232                           BGE     2$              ;Loop for each row of A
3753                                        ;+
3754                                        ; Now rescale the result
3755                                        ;-
3756 003176  013700  003140'                  MOV     RESPTR,R0       ;Point to the result
3757 003202  013701  005070'                  MOV     OPTMP1,R1       ;Get I
3758 003206  013703  005074'                  MOV     OPTMP3,R3       ;Point to Factor
3759 003212                           9$:     LDF     (R3)+,AC1       ;Get FACTOR(i)
3760 003220  013702  005072'                  MOV     OPTMP2,R2       ;Get J
3761 003224                           10$:    LDF     (R0),AC0        ;Get Z(i,j)
3762 003232                                   DIVF    AC1,AC0         ;Z(i,j) / FACTOR(i)
3763 003240                                   STF     AC0,(R0)+       ;Store the scaled result
Z       003244  010020                        MOV AC0,(R0)+
3764 003246                                   SOB     R2,10$          ;Loop for each column
3765 003252                                   SOB     R1,9$           ;Loop for each row
3766
3767 003256                                   CALL    KILL            ;Delete FACTOR and PP vectors
3768 003262                                   RETURN                  ;All done at last
3769                                        ;+
3770                                        ; This subroutine performs the inner product between various column vectors
3771                                        ; pointed to by the following register setup.
3772                                        ;       R0 - Number of elements in vectors
3773                                        ;       R1 - Pointer to first vector
3774                                        ;       R2 - Pointer to second vector
3775                                        ;       R3 - Step to next element for vector 1
3776                                        ;       R4 - Step to next element for vector 2
3777                                        ;       AC0 - Inner product
3778                                        ;       AC1 - Temporary
3779                                        ;       AC2 - Temporary
3780                                        ;-
3781
3782 003264                           INNER:  SETD                    ;Force double precision mode
3783 003270                                   CLRF    AC0             ;Zero the result
3784 003276  023727  000062' 000004   1$:     CMP     NUMLEN,#4       ;Are we in single precision mode?
3785 003304  001007                           BNE     2$              ;No, good
3786 003306                                   LDCFD   (R1),AC1        ;Convert element of vector 1 to double precision
3787 003314                                   LDCFD   (R2),AC2        ;Convert element of vector 2 to double precision
3788 003322  000406                           BR      3$
3789 003324                           2$:     LDF     (R1),AC1        ;Get next element of vector 1
3790 003332                                   LDF     (R2),AC2        ;Get next element of vector 2
3791 003340                           3$:     MULF    AC2,AC1         ;Form the product
3792 003346                                   ADDF    AC1,AC0         ;Accumulate the sum
3793 003354  060301                           ADD     R3,R1           ;Next element of vector 1
3794 003356  060402                           ADD     R4,R2           ;Next element of vector 2
3795 003360                                   SOB     R0,1$           ;Loop for all the elements
3796 003364  023727  000062' 000004           CMP     NUMLEN,#4       ;Are we in single precision mode?
3797 003372  001005                           BNE     4$              ;No
3798 003374                                   SETF                    ;Yes, Change back to single precision mode
3799 003400                                   LDCDF   AC0,AC0         ;Convert the result back too
3800 003406                           4$:     RETURN                  ;Done
3801
3802
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 66-3
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

                                        ;+
                                        ; This subroutine applies the Householder transformations to the array
                                        ; pointed to by OPTMP9.  Other inputs include:
                                        ;       OPTMP8 - Initial column
                                        ;       Top of stack - Number of columns to process
                                        ;       R4 - Inner product increment for input columns
                                        ; The value on the top of the stack is left there by TRANS and must be
                                        ; popped off after TRANS has returned.
                                        ;-

003410  005237  005106'         TRANS:  INC     OPTMP8          ;J = J + 1
003414  023766  005106' 000002          CMP     OPTMP8,2(SP)    ;Any columns left?
003422  002401                          BLT     1$              ;Yes
003424                                  RETURN                  ;No, Then all done
003426  013700  005066'         1$:     MOV     OPTMP0,R0       ;Get M
003432  163700  005112'                 SUB     OPTM10,R0       ;The number of elements in product
003436  016601  000004                  MOV     4(SP),R1        ;Pointer to K-th column of A
003442  013702  005110'                 MOV     OPTMP9,R2       ;Get pointer to J-th column
003446                                  CALL    INNER           ;Get inner product of Kth col of A anc Jth col
003452                                  DIVF    AC4,AC0         ;MV = A(k,k).A(k,j) / T
003460  013700  005066'                 MOV     OPTMP0,R0       ;Get M
003464  163700  005112'                 SUB     OPTM10,R0       ;The number of elements to loop on
003470  016601  000004                  MOV     4(SP),R1        ;Point to K-th column
003474  013702  005110'                 MOV     OPTMP9,R2       ;Point to J-th column
003500                          2$:     LDF     (R1),AC1        ;Get A(i,k)
003506                                  MULF    AC0,AC1         ;A(i,k) * MV
003514                                  LDF     (R2),AC2        ;Get A(i,j)
003522                                  SUBF    AC1,AC2         ;A(i,j) - A(i,k) * MV
003530                                  STF     AC2,(R2)        ;A(i,j) = A(i,j) - A(i,k) * MV
003536  060301                          ADD     R3,R1           ;Next row of column K
003540  060402                          ADD     R4,R2           ;Next row of column J
003542                                  SOB     R0,2$           ;Loop on each row
003546  063737  000062' 005110'         ADD     NUMLEN,OPTMP9   ;Next column of second array
003554  000715                          BR      TRANS           ;Loop on each column

                                ;+
                                ; This subroutine is used to swap rows or columns of a matrix.  The input
                                ; parameters are:
                                ;       R0 - The number of elements to swap
                                ;       R1 - Index of first row or column
                                ;       R2 - Offset for converting index to a pointer
                                ;       R3 - Not an input, but used to hold the index K
                                ;       R4 - Increment for stepping along row or column
                                ;       2(SP) - Pointer to matrix whose rows or columns are being swapped
                                ;               This must be removed when finished using the routine.
                                ;-

003556                          SWAP:   MUL     R2,R1           ;Convert index to an offset
003570  066601  000002                  ADD     2(SP),R1        ;Convert offset to a pointer
003574  013703  005112'                 MOV     OPTM10,R3       ;Get index, K
003600                                  MUL     R2,R3           ;Convert it to an offset
003612  066603  000002                  ADD     2(SP),R3        ;Convert the offset to a pointer
003616                          1$:     LDF     (R1),AC0        ;Get element of first row or column
003624                                  LDF     (R3),AC1        ;Get element of K-th row or column
003632                                  STF     AC0,(R3)        ;Swap them
003640                                  STF     AC1,(R1)        ; around
003646  060401                          ADD     R4,R1           ;Next element of first row or column
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 66-4
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   3860 003650  060403                          ADD     R4,R3           ;Next element of K-th row or column
   3861 003652                                   SOB     R0,1$           ;Loop for all the elements of the row or column
   3862 003656                                   RETURN
   3864 003660                                   LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 67
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   3866 003660                          DOPINS  MEM,70
   3921 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 68
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   3923 003660                          MOPINS  NUB,106
   3986 003660                                  LIST
   3987
   3988
   3989
   3990
   3991
   3992 003660                          DOPINS  UNON,106
   4022 003660                                  LIST
   4023
   4024
   4025
   4026
   4027
   4028 003660                          DOPINS  INTR,66
   4042 003660                                  LIST
   4043
   4044
   4045
   4046
   4047
   4048 003660                          DOPINS  SDIF,50
   4161 003660                                  LIST
```

I 11

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 69
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   4163 003660                          DOPINS  SCON,234
   4179 003660                                  LIST
   4180
   4181
   4182 003660                          DOPINS  CONT,112
   4200 003660                                  LIST
   4201
   4202
   4203 003660                          DOPINS  PSUB,236
   4209 003660                                  LIST
   4210
   4211
   4212 003660                          DOPINS  SUBS,116
   4288 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 70
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   4290 003660                          MOPINS  EXE,70
   4336 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 71
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   4338 003660                          MOPINS  ENCD,76
   4411 003660                                  LIST
```

```
 4413 003660                            MOPINS  FMT,160
 4423 003660                                    LIST
 4424
 4425
 4426
 4427
 4428
 4429 003660                            DOPINS  FMT,160
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 73
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   4953 003660                                  LIST
```

```
APLOU8  MACRO M1113  08-FEB-80 10:28  PAGE 74
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   4955 003660                          MOPINS  IBM,132,,,MONIBM
   4958 003660                                  LIST
   4959
   4960
   4961 003660                          DOPINS  IBM,132
   5046 003660                                  LIST
```

```
 5048 003660                            MOPINS  QUAD,74,T.NIL+T.SPOP
 5075 003660                                    LIST
 5076
 5077
 5078
 5079
 5080
 5081 003660                            DOPINS  QUAD,74,T.NIL+T.SPOP
 5097 003660                                    LIST
 5098
 5099
 5100
 5101
 5102
 5103 003660                            MOPINS  QQAD,142,T.NIL+T.SPOP
 5135 003660                                    LIST
 5136
 5137
 5138
 5139
 5140
 5141 003660                            DOPINS  QQAD,142,T.NIL+T.SPOP
 5152 003660                                    LIST
 5153
 5154
 5155
 5156
 5157
 5158 003660                            MOPINS  DQAD,134,T.NIL+T.SPOP
 5192 003660                                    LIST
 5193
 5194
 5195
 5196 003660                            DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
 5199 003660                                    LIST
```

```
5201 003660                          MOPINS  BXAV,162,T.NIL
5219 003660                                  LIST
5220
5221
5222
5223
5224
5225 003660                          MOPINS  BXCT,164,T.NIL+T.SPOP
5234 003660                                  LIST
5235
5236
5237 003660                          DOPINS  BXCT,164,T.NIL+T.SPOP
5272 003660                                  LIST
5273
5274
5275
5276
5277
5278 003660                          MOPINS  BXIO,166,T.NIL+T.SPOP
5287 003660                                  LIST
5288
5289
5290 003660                          DOPINS  BXIC,166,T.NIL+T.SPOP
5302 003660                                  LIST
5303
5304
5305
5306
5307 003660                          MOPINS  BXLC,212,T.NIL
5340 003660                                  LIST
5341
5342
5343
5344
5345
5346
5347 003660                          MOPINS  BXPP,170,T.NIL+T.SPOP
5356 003660                                  LIST
5357
5358
5359 003660                          DOPINS  BXPP,170,T.NIL+T.SPOP
5381 003660                                  LIST
5382
5383
5384
5385
5386
5387
5388 003660                          MOPINS  BXPW,172,T.NIL+T.SPOP
5397 003660                                  LIST
5398
5399
5400 003660                          DOPINS  BXPW,172,T.NIL+T.SPOP
5417 003660                                  LIST
5418
5419
5420
```

```
APLOO8  MACRO M1113  08-FEB-80 10:28  PAGE 76-1
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5421
   5422
   5423 003660                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 003660                                  LIST
   5433
   5434
   5435 003660                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 003660                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 003660                          MOPINS  BXTS,200,T.NIL
   5562 003660                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 003660                          MOPINS  BXWA,176,T.NIL
   5592 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 77
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5594 003660                          MOPINS  BXCR,206
   5716 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 78
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5718 003660                          MOPINS  BXEX,202
   5775 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 79
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5777 003660                          MOPINS  BXFX,210
   5879 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 80
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5881 003660                          MOPINS  BXNC,204
   5950 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 81
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   5952 003660                          MOPINS  BXNL,214
   5963 003660                                  LIST
   5964
   5965
   5966 003660                          DOPINS  BXNL,214
   6104 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 82
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   6106 003660                          DOPINS  FGET,150,T.DIM
   6330 003660                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 003660                          DOPINS  FPUT,152,T.DIM
   6485 003660                                  LIST
```

L 12

```
APLOO8  MACRO M1113  08-FEB-80 10:28  PAGE 83
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   6487 003660                          DOPINS  FPNT,154,T.DIM
   6714 003660                                  LIST
```

M 13

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 84
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   6716 003660                          MOPINS  1SCN,130
   6723 003660                                  LIST
   6724
   6725
   6726 003660                          MOPINS  SCAN,62,T.DIM
   6732 003660                                  LIST
   6733
   6734
   6735 003660                          MOPINS  1RDT,124
   6741 003660                                  LIST
   6742
   6743
   6744 003660                          MOPINS  RDCT,60,T.DIM
   7044 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 85
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

    7046 003660                          MOPINS  GETIDN,0
    7193 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 86
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   7195 003660                          MOPINS  SATAN,0
   7404 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 87
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

    7406 003660                          MOPINS  OUTR,156
    7476 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 88
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   7478 003660                          MOPINS  DOUTER,0
   7504 003660                                  LIST
   7505
   7506
   7507
   7508
   7509
   7510 003660                          MOPINS  PRDSET,0
   7564 003660                                  LIST
   7565
   7566
   7567
   7568
   7569
   7570 003660                          MOPINS  INSPAC,0
   7612 003660                                  LIST
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 89
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   7614 003660                          DOPINS  INNR,156
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 90
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   7960 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 91
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   7962 003660                          MOPINS  SUBR,0
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 92
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   8725 003660                                    LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 93
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

   8727 003660                          MOPINS  RAND,0
   8771 003660                                  LIST
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 94
MIXED FUNCTION          MATRIX DIVIDE (LEAST SQUARES FIT)

      1         000001                              .END
```

```
APLO08  MACRO M1113  08-FEB-80 10:28  PAGE 94-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.ALOC= 000162  D.CTLM= 000000  EDIFLG  005062R  025
AC1    =%000001          CHRLBR= 000073  C.BLOK= 000064  D.DEAL= 000003  EDIT   = 000003
AC2    =%000002          CHRLPR= 000072  C.BLSZ= 000140  D.DECD= 000002  EIS    = 000000
AC3    =%000003          CHRMNS= 000137  C.BUFF= 177760  D.DIV = 000000  ELESIZ  005132R  025
AC4    =%000004          CHRMUL= 000075  C.BUFL= 177762  D.DMD = 000010  EMPTY = ****** GX
AC5    =%000005          CHRNEG= 000100  C.CCNT= 177766  D.DQAD= 177777  ENDCOR  000004R  025
ALLINT  003274R  025     CHRPDL= 000271  C.CPTR= 177764  D.DROP= 000002  EOF    = 104475
ARYRET  003276R  025     CHRPLU= 000055  C.CZ  = 001000  D.ENCD= 000002  ERRMAX= 000166
ASCQUO= 000042           CHRPWR= 000120  C.DIG = 000040  D.EQ  = 000000  ERRNUM  004772R  025
A.ALEN= 000004           CHRQST= 000121  C.FASW= 020000  D.EXPD= 000002  ERRPTR  004770R  025
A.ARG = 000040           CHRQUO= 000113  C.FLEN= 000010  D.FGET= 000005  ERRO    004774R  025
A.ASB = 000200           CHRRBR= 000047  C.FLSH= 004000  D.FMT = 000004  EXEBUF  005054R  025
A.BITS= 000003           CHRRPR= 000042  C.FRC = 000001  D.FPNT= 000005  E$CT    005056R  025
A.CON = 000020           CHRSEM= 000074  C.IDEN= 000010  D.FPUT= 000005  E$PT    005060R  025
A.DIMS= 000002           CHRSMO= 000112  C.NAME= 000102  D.GE  = 000000  FA.APD= 000100
A.DIM1= 000006           CHRSPA= 000040  C.NOBF= 000400  D.GT  = 000000  FA.CRE= 000010
A.NUM = 000006           CHRUPA= 000131  C.NUM = 000020  D.IBM = 000002  FA.DLK= 001000
A.PTR = 000000           CHR.BS= 000010  C.OP  = 000004  D.IND = 000002  FA.ENB= 100000
A.TMP = 000010           CHR.CC= 000003  C.OPN = 010000  D.INNR= 000013  FA.EXC= 002000
BACK    002520R  020     CHR.CR= 000015  C.RDLN= 177776  D.INTR= 000006  FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032  C.REWT= 002000  D.LE  = 000000  FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046  C.RND = 000010  D.LOG = 000001  FA.NSR= 000200
BLZBUB  005116R  025     CHR.LF= 000012  C.SBIT= 177756  D.LT  = 000000  FA.POS= 010000
BRSELF  005046R  025     CHR.LT= 000043  C.SPFX= 000100  D.MAX = 000000  FA.RD  = 000001
CCNT    005072R  025     CHR.SI= 000017  C.SPGL= 000200  D.MEM = 000002  FA.RWD= 004000
CHANEX  003060R  025     CHR.SL= 000057  C.SPLX= 000001  D.MIN = 000000  FA.SEQ= 040000
CHAN00  000256R  025     CHR.SO= 000016  C.SPRS= 000002  D.MUL = 000000  FA.SHR= 000040
CHAN01  000330R  025     CHR.TB= 000011  C.SRC = 177774  D.NE  = 000000  FA.TMP= 000020
CHAN02  000512R  025     CHR.UL= 000106  C.SRCP= 000024  D.NN  = 000000  FA.WCK= 020000
CHAN03  000674R  025     CHR..A= 000141  C.TT  = 000004  D.NR  = 000000  FA.WRT= 000002
CHAN04  001056R  025     CHR..E= 000145  C.WIDC= 177770  D.OR  = 000000  FD.BLK= 000010
CHAN05  001240R  025     CHR..S= 000163  C.WIDR= 177772  D.PLUS= 000000  FD.CCL= 000002
CHAN06  001422R  025     CHR..T= 000164  DEBUG = 000000  D.PSUB= 000006  FD.COM= 020000
CHAN07  001604R  025     CHR..U= 000165  DEFERR= 104405  D.PWR = 000001  FD.CR  = 000002
CHAN08  001766R  025     CHR..Z= 000172  DEVIL   005114R  025 D.QQAD= 000000  FD.DIR= 000010
CHAN09  002150R  025     CHR..0= 000060  DEVNTF= 104470  D.QUAD= 000000  FD.FTN= 000001
CHAN10  002332R  025     CHR..1= 000061  DIGITS  000006R  025 D.RES = 000000  FD.F11= 040000
CHAN11  002514R  025     CH.AND= 000001  DIMAX = 000020  D.RHO = 000000  FD.INS= 000010
CHAN12  002676R  025     CMDBUF  003250R  025 DIMBLK  003112R  025 D.ROT = 000003  FD.ISP= 002000
CHAN13  000022R  023     CMDMAX= 000010  DIMPTR  003114R  025 D.SCON= 000006  FD.MNT= 100000
CHAN14  000204R  023     CNTSIZ  005130R  025 DIMTYP  003116R  025 D.SDIF= 000006  FD.OSP= 004000
CHNOPN= 104471           COMBIN  005076R  025 DIRFUL= 104466  D.SUB = 000000  FD.PLC= 000004
CHRAND= 000051           COMCNT  005126R  025 DOMERR= 104417  D.SUBS= 000006  FD.PRN= 000004
CHRBEL= 000007           COMERR= 104425  DPTERR= 104416  D.TAKE= 000002  FD.PSE= 010000
CHRBKR= 000133           COMPTR  005122R  025 DYDCNT  003224R  025 D.TRAN= 000004  FD.RAH= 000001
CHRBOL= 000200           COMRTN  003156R  025 D$DMD   000460RG  020 D.UNON= 000006  FD.RAN= 000002
CHRBOX= 000114           COMSKP  005124R  025 D.AND = 000000  D.$MAX= 000000  FD.REC= 000001
CHRCMA= 000054           CONVER= ****** GX  D.BXCT= 000007  D.$MIN= 000000  FD.RWM= 000001
CHRCOL= 000076           CORPTR  005120R  025 D.BXIO= 000007  D.$MUL= 000000  FD.SDI= 000020
CHRDEL= 000107           CS.ASS= 000200  D.BXNL= 000007  D.$PLU= 000000  FD.SQD= 000040
CHRDIV= 000053           CS.BOO= 000010  D.BXPP= 000007  D.$RES= 000000  FD.TTY= 000004
CHRDLR= 000176           CS.FLO= 000020  D.BXPW= 000007  D.$SUB= 000000  FD.WBH= 000002
CHRDLT= 000110           CS.IDN= 000040  D.BXRL= 000007  D.1COM= 000001  FF.CHR= 000005
CHRDOT= 000056           CS.ONE= 000100  D.CIRC= 000001  D.1EXP= 000002  FF.NV  = 000003
CHREP  = 000105          CTEPS   000152R  025 D.COMB= 000001  D.1ROT= 000003  FF.POE= 000002
CHRERR= 104431           CTMAX   000212R  025 D.COMP= 000001  EDICHR  005063R  025 FF.RWD= 000001
CHRFIL= 000177           CTRMAX  000172R  025 D.CONT= 000006  EDICNT  005064R  025 FF.RWF= 000006
```

APLOO8 MACRO M1113 08-FEB-80 10:28 PAGE 94-2
SYMBOL TABLE

```
FF.SPC= 000004          F.ERR = 000052
FHEADR  005012R     025 F.FACC= 000043
FHRET   005040R     025 F.FFBY= 000014
FILNTF= 104467          F.FLAG= 000002
FIS   = 000000          F.FLB = 000100
FISCON= 046000          F.FNAM= 000110
FIXPTR  003074R     025 F.FNB = 000102
FLAGS   004764R     025 F.FNDF= 000010
FLARGC= ****** GX       F.FTYP= 000116
FLCBUF  003250R     025 F.FUN = 000010
FLINE   005014R     025 F.FVER= 000120
FLOEXP  003270R     025 F.HEAD= 000006
FLONEG  003272R     025 F.HIBK= 000004
FLSWTB  000222R     025 F.LBL = 000020
FLTLEN= 000010          F.LINE= 000012
FMONCT= ****** GX       F.LOC = 000004
FNINSI  005044R     025 F.LSTC= 000004
FORIGN  000032R     025 F.LUN = 000042
FO.APD= 000106          F.MBCT= 000054
FO.MFY= 000002          F.MBC1= 000055
FO.RD = 000001          F.MBFG= 000056
FO.UPD= 000006          F.MON = 000002
FO.WRT= 000016          F.NIL = 000001
FPMODE  000042R     025 F.NRBD= 000024
FPP   = 000000          F.NREC= 000030
FPPLEN= 000004          F.OPN = 000010
FRARGC= ****** GX       F.OVBS= 000030
FREBLK  007400R     025 F.PREV= 000000
FUZZ    000142R     025 F.PRNT= 100000
FXPTR1  003076R     025 F.PROT= 010000
F$EVAL= 000100          F.RACC= 000016
F$EVI = 000200          F.RATT= 000001
F$EXEC= 000400          F.RCNM= 000034
F$FUNC= 001000          F.RCTL= 000017
F.ACTL= 000076          F.READ= 004000
F.ALOC= 000040          F.REDF= 000040
F.BBFS= 000062          F.RES = 000040
F.BDB = 000070          F.RET = 177772
F.BGBC= 000057          F.RSIZ= 000002
F.BITS= 177774          F.RTYP= 000000
F.BKDN= 000026          F.R2  = 000014
F.BKDS= 000020          F.R3  = 000016
F.BKEF= 000050          F.R4  = 000020
F.BKP1= 000051          F.SEQN= 000100
F.BKST= 000024          F.SPDV= 000072
F.BKVB= 000064          F.SPUN= 000074
F.BR  = 000020          F.STBK= 000036
F.CHR = 000075          F.STK = 000022
F.CNTG= 000034          F.STOP= 000004
F.CODE= 000010          F.STP = 000001
F.DFNB= 000046          F.SUSP= 000002
F.DSPT= 000044          F.TRC = 000002
F.DVNM= 000134          F.UNIT= 000136
F.EFBK= 000010          F.URBD= 000020
F.EFN = 000050          F.VAL = 177776
F.END = 002000          F.VBN = 000064
F.EOBB= 000032          F.VBSZ= 000060

GETFRA= ****** GX
GNOME   003102R     025 L.CNT = 000002
HASHC   005136R     025 L.PRNT= 000001
HASHTB  005140R     025 L.PTR = 000000
HOUSE   001632R     020 L.PTR1= 000004
HSHMSK= 177601          L.SUBS= 000002
HSHSIZ= 000200          MACHIN= 000106
IASH  = ****** GX       MAXCHN= 000014
ICHAN   005050R     025 MONBLK  003120R     025
ILFNAM= 104464          MONCVT  003132R     025
ILLCHN= 104540          MONLEN  003130R     025
ILLSW = 104565          MONPTR  003122R     025
IMUL  = ****** GX       MONSFT  003126R     025
INDERR= 104410          MONTYP  003124R     025
INNER   003264R     020 MONVAL  003134R     025
INNTMP  003250R     025 MSMMD   000000RG    020
INNTM2  005016R     025 M.ABS   000000
INPTR   000064R     025 M.BOL   000200
INSNAM= 104462          M.BXAV= 000007
IOERR = 104477          M.BXCR= 000007
I.CNT = 177776          M.BXCT= 000007
I.DIM = 177772          M.BXEX= 000007
I.LEN = 000012          M.BXFX= 000007
I.OERR= 000062          M.BXIO= 000007
I.OLD = 177770          M.BXLC= 000003
I.PTR = 177774          M.BXNC= 000007
I.STR = 177766          M.BXNL= 000007
KILL  = ****** GX       M.BXPP= 000007
LABERR= 104406          M.BXPW= 000007
LABLPT  005042R     025 M.BXRL= 000007
LARGBL  003162R     025 M.BXTS= 000007
LARGCV  003176R     025 M.BXWA= 000007
LARGFL  003160R     025 M.CEI = 177777
LARGLE  003174R     025 M.CHR = 000202
LARGLN  003166R     025 M.CMT = 000206
LARGPT  003164R     025 M.DOUT= 000006
LARGSF  003172R     025 M.DQAD= 000006
LARGTY  003170R     025 M.ENCD= 000004
LARGVA  003200R     025 M.EXE = 000000
LBINC   005100R     025 M.EXP = 000001
LBUFF   003316R     025 M.FACT= 000001
LBUFLN= 000621          M.FIL = 000177
LCOUNT  005106R     025 M.FLR = 177777
LENERR= 104412          M.FMT = 000004
LENTAB  000054R     025 M.GDUP= 000003
LEXPTR  003100R     025 M.GDWN= 000003
LFTOP   005066R     025 M.GETI= 000006
LINC    005102R     025 M.IBM = 000003
LINE    005016R     025 M.IDN = 000000
LININC  005026R     025 M.INSP= 000006
LISIZE= 000020          M.IOTA= 000000
LITOP   005072R     025 M.LIT = 000205
LNLNG = 104541          M.LN  = 000001
LOCLCT  005036R     025 M.MMD = 000010
LOWCOR  000002R     025 M.NEG = 000000
LSTERP  004766R     025 M.NOT = 000000
LWIDTH  000070R     025 M.NUB = 000006

M.NUM = 000204
M.OUTR= 000012
M.PITM= 000000
M.PRDS= 000006
M.QQAD= 000000
M.QUAD= 000000
M.RAND= 000012
M.RAVL= 000000
M.RCIP= 000000
M.RDCT= 000011
M.REVS= 000003
M.ROLL= 000000
M.SATA= 000001
M.SCAN= 000011
M.SHAP= 000000
M.SIGN= 000000
M.SUBR= 000001
M.SYM = 000203
M.SYN = 000201
M.TRAN= 000012
M.$ABS= 000000
M.$CEI= 000000
M.$FLR= 000000
M.$IDN= 000000
M.$NEG= 000000
M.$SIG= 000000
M.1RDT= 000011
M.1REV= 000003
M.1SCN= 000011
NAMBUF  003240R     025
NB.DEV= 000200
NB.DIR= 000100
NB.NAM= 000004
NB.SD1= 000400
NB.SD2= 001000
NB.SNM= 000040
NB.STP= 000020
NB.SVR= 000010
NB.TYP= 000002
NB.VER= 000001
ND.DEV= 000200
ND.EXT= 000002
ND.NAM= 000004
ND.PPN= 000100
ND.$EX= 000000
ND.$ES= 000020
ND.$NM= 000000
ND.$NS= 000040
ND.$PN= 001400
NEGSGN  005063R     025
NINPT1  003302R     025
NINPT2  003300R     025
NINSTK  003304R     025
NINTMP  003310R     025
NINTYP  003306R     025
NOTAVL= 104472
NOTOPN= 104473
```

```
APLOO8  MACRO M1113  08-FEB-80 10:28  PAGE 94-3
SYMBOL TABLE

NPRT1    004762R  025 OVROPR= 000010          R.OFF  = 000000          SY.SRT= 000400          UPPER  = 177400
NULL     007364R  025 PATCH   007344R  025 R.PREV= 000004          SY.SVC= 000004          UTNAME  000344R  023
NUMCHN=  000020       PRECSN  000052R  025 R.PTR  = 000000          SY.SVL= 000200          VALERR= 104413
NUMDFL   003230R  025 RANDAC  003064R  025 R.SEQ  = 000003          S.AB   = 177776          VARBLE  007400R  025
NUMEFL   003233R  025 RANDY   003062R  025 R.VAR  = 000002          S.ASB  = 100000          VARBLS  000000R  025
NUMESN   003236R  025 RANERR= 104411          R6     =%000006          S.BFHD= 000020          VERNUM  000000R  025
NUMFLG   003230R  025 RARGBL  003204R  025 R7     =%000007          S.BITS= 000002          VRSION= 000002
NUMLEN   000062R  025 RARGCV  003220R  025 SI     =%000005          S.CHR1= 000003          WIDTH   000072R  025
NUMNEG   003232R  025 RARGFL  003202R  025 SIDAMN= 104433          S.FATT= 000016          WROUND  000020R  025
NUMQFL   003234R  025 RARGLE  003216R  025 SIFRAM  003314R  025 S.FDB  = 000140          WSDATE  000136R  025
NUMRIG   003231R  025 RARGLN  003210R  025 SISTAK  006350R  025 S.FLB  = 040000          WSFULL= 104403
NUMWID   003226R  025 RARGPT  003206R  025 SITOP   006346R  025 S.FNAM= 000006          WSNAME  000074R  025
NUMZFL   003235R  025 RARGSF  003214R  025 SITP    007144R  025 S.FNB  = 000036          WSTIME  000132R  025
NYIERR=  104432       RARGTY  003212R  025 SIWARN= 000200          S.FNBW= 000017          $ABSF  = ****** GX
N.BLSZ=  000036       RARGVA  003222R  025 SI7TAB  000044R  025 S.FNTY= 000004          $ADDF  = ****** GX
N.DESC=  000020       RBASE   005074R  025 SI.BIT= 177774          S.FTYP= 000002          $CFCC  = ****** GX
N.DEV =  000032       RCOUNR  005112R  025 SI.LEN= 177772          S.LEN  = 000002          $CLRF  = ****** GX
N.DEVN=  000034       RCOUNT  005110R  025 SI.RET= 177772          S.LINK= 177774          $CMPF  = ****** GX
N.DID =  000024       REDCNT  005070R  025 SI.VAL= 177776          S.NFEN= 000020          $DIVF  = ****** GX
N.DVNM=  000032       REDIGN  005066R  025 SKPSIZ  005134R  025 S.TYPE= 140000          $FIS   = ****** G
N.EXT =  000014       REDOFF  005072R  025 SPHOLD  000004R  024 S.VAL  = 000000          $FPP   = ****** G
N.FID =  000000       REENTR  003060R  025 SPINIT  000002R  024 TKDIMS  005072R  025 $FPPLN= ****** G
N.FNAM=  000006       REGTYP= 000003          STKINC= 000774          TKFRAM  005066R  025 $LDCDF= ****** GX
N.FTYP=  000014       RESBIT  003142R  025 STOP   = 104420          TKLEFT  005070R  025 $LDCFD= ****** GX
N.FVER=  000016       RESCVT  003152R  025 SUBCNT  003104R  025 TMODE   000403R  023 $LDF   = ****** GX
N.NAME=  000006       RESERR= 104440          SUBDCT  003106R  025 TMPTAG= 000166          $MACHN= ****** G
N.NEXT=  000022       RESLEN  003150R  025 SUBPTR  003110R  025 TRANS   003410R  020 $MULF  = ****** GX
N.PPN =  000024       RESPTR  003140R  025 SWAP    003556R  020 TTYBUF  004140R  025 $NEGF  = ****** GX
N.STAT=  000020       RESSFT  003146R  025 SYMB1   005344R  025 TTYLEN= 000622          $RSTS  = ****** G
N.UNIT=  000034       RESTYP  003144R  025 SYMF    005340R  025 TURKEY= 104436          $RSX   = ****** G
OBUFLN=  001000       RESULT  003136R  025 SYMPT1  005346R  025 TYPE   = 000166          $RT11  = ****** G
OCHAN    005052R  025 RESVAL  003154R  025 SYMTAB  005352R  025 T.BOL  = 020000          $SETD  = ****** GX
OMODE    000402R  023 RINC    005104R  025 SYMTBF= 104404          T.CON  = 004000          $SETF  = ****** GX
OPTMP0   005066R  025 RLTBAS  005066R  025 SYMTOP  005350R  025 T.DIM  = 000100          $STF   = ****** GX
OPTMP1   005070R  025 ROTINC  005070R  025 SYMTP   006272R  025 T.FUN  = 010000          $SUBF  = ****** GX
OPTMP2   005072R  025 ROUND   000010R  025 SYMWRN= 000054          T.LIST= 000400          $TSTF  = ****** GX
OPTMP3   005074R  025 ROVER   003312R  025 SYNERR= 104407          T.LIT  = 000020          .BOOLE= 000002
OPTMP4   005076R  025 RSTS   = 000000          SYSERR= 104402          T.MON  = 000010          .CHARA= 000000
OPTMP5   005100R  025 RSX    = 000115          SYSTAT  000000R  024 T.NIL  = 000004          .DEF. = 000002
OPTMP6   005102R  025 RTOP     005070R  025 SY.BAD= 000001          T.NUM  = 002000          .FLOAT= 000006
OPTMP7   005104R  025 RT11   = 000000          SY.BRK= 100000          T.OPFN= 000001          .FSRCB= ****** G
OPTMP8   005106R  025 R.BITS= 000003          SY.BS  = 000010          T.OTHR= 040000          .INTEG= 000004
OPTMP9   005110R  025 R.FIX  = 000001          SY.CLR= 000040          T.SOP  = 000002          .MOLUN  000404R  023
OPTM10   005112R  025 R.LCT  = 000002          SY.ERP= 000020          T.SPOP= 000040          .SQRT  = ****** GX
OPTM14   005026R  025 R.LN   = 000002          SY.NOU= 000100          T.TMP  = 000200          ...GBL= 000000
ORIGIN   000030R  025 R.LNUM= 000010          SY.OPN= 000002          T.VAL  = 100000          ...TPC= 000020
OUTPTR   000066R  025 R.NEXT= 000006          SY.PMT= 001000          T.VAR  = 001000

. ABS.   000000  000
         000000  001
TABLES   000000  002
MONTB    000000  003
DYDTB    000000  004
BOPTB    000000  005
MONBTB   000000  006
DYDBTB   000000  007
COMTB    000000  010
```

```
APL008  MACRO M1113  08-FEB-80 10:28  PAGE 94-4
SYMBOL TABLE

CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    003660      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  13551 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:33
APL008,DM1:[305,301]APL008/-SP=APLRX1,APLDEF,APL008,APLOP,END
```

FILEID**APL009

```
  AAAAAA      PPPPPPPP      LL              000000      000000      999999
  AAAAAA      PPPPPPPP      LL              000000      000000      999999
AA      AA    PP      PP    LL            00      00  00      00  99      99
AA      AA    PP      PP    LL            00      00  00      00  99      99
AA      AA    PP      PP    LL            00      00  00    0000  99      99
AA      AA    PP      PP    LL            00      00  00    0000  99      99
AA      AA    PPPPPPPP      LL            00      00  00  00  00    99999999
AA      AA    PPPPPPPP      LL            00      00  00  00  00    99999999
AAAAAAAAAA    PP            LL            00      00  0000    00          99
AAAAAAAAAA    PP            LL            00      00  0000    00          99
AA      AA    PP            LL            00      00  00      00          99
AA      AA    PP            LL            00      00  00      00          99
AA      AA    PP            LLLLLLLLLL      000000      000000      999999
AA      AA    PP            LLLLLLLLLL      000000      000000      999999
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 10:31
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    84- 6717    OPERATORS                FIRST COORDINATE SCAN
```

APL009 MACRO M1113 08-FEB-80 10:31
TABLE OF CONTENTS

84- 6727 OPERATORS SCAN
84- 6736 OPERATORS FIRST COORDINATE REDUCTION
84- 6745 OPERATORS REDUCTION

```
APLO09  MACRO M1113  08-FEB-80 10:31  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APL009
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APL009 IS:
      9                                 ;
     10                                 ;       APL009 < ...,APL009,APLOP
     11                                 ;
     12
     13          000011                OVROPR= 9.                      ;OVERLAY SPECIFIED OPERATORS
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 46
DEFINITIONS     ERRORS

      5
      6                                 ;++
      7                                 ; FUNCTIONAL DESCRIPTION:
      8                                 ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
      9                                 ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
     10                                 ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
     11                                 ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
     12                                 ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
     13                                 ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
     14                                 ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
     15                                 ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
     16                                 ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
     17                                 ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
     18                                 ;       INTACT!
     19                                 ;--
     24
     27
    124
    125                                 ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
    126                                 ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
    127                                 ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
    128                                 ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
    129                                 ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
    130                                 ;MACROS.
    131
    132                                 .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
    133                                 .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
    134                                 .IF EQ OVROPR-D.'NAME
    135                                         .PSECT  CODE
    136                                 D$'NAME::
    137                                 .ENDM
    138
    139                                 .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
    140                                 .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
    141                                 .IF EQ OVROPR-M.'NAME
    142                                         .PSECT  CODE
    143                                         .IF NE NUM
    144                                 M$'NAME::
    145                                     .ENDC
    146                                 .ENDM
    147
    148                                 .MACRO  LIST
    149                                         .ENDC
    150                                 .ENDM
    151
    152                                 .MACRO  PRMRES
    153                                         .IF EQ 1
    154                                 .ENDM
    155
    159                                         ;************ ASSEMBLING THE FILE!!!!!
    160 007400                          PRMRES
    199 007400                                  LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,.INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,.INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,.INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,.FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 47-1
DEFINITIONS     ERRORS

    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

```
641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
780 007400                                  LIST
781
```

DEFINITIONS   ERRORS

```
783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
786 007400                                  LIST
787
788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
791 007400                                  LIST
792
793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
799 007400                                  LIST
800
801
802 007400                          MOPINS  $IDN,-2
808 007400                                  LIST
809
810
811 007400                          MOPINS  $FLR,-14
855 007400                                  LIST
856
857
858 007400                          MOPINS  $CEI,-16
868 007400                                  LIST
869
870
871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
878 007400                                  LIST
879
880 007400                          MOPINS  $NEG,-4
887 007400                                  LIST
888
889
890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
901 007400                                  LIST
902
903 007400                          MOPINS  $SIG,-6
914 007400                                  LIST
915
916
917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
928 007400                                  LIST
929
930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
939 007400                                  LIST
940
941
942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
949 007400                                  LIST
950
951
952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
959 007400                                  LIST
960
961 007400                          PRMRES
966 007400                                  LIST
967
968
969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
977 007400                                  LIST
978
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,,FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   1189 007400                          MOPINS  SHAP,102
   1211 007400                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

```
APLO09  MACRO M1113  08-FEB-80 10:31  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

```
APLO09  MACRO M1113  08-FEB-80 10:31  PAGE 54
DEFINITIONS     ERRORS

   1701 007400                          DOPINS  1COM,124
   1709 007400                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 007400                          DOPINS  COMP,60,T.DIM
   1789 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2206 007400                          DOPINS  1ROT,146
   2214 007400                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 007400                          DOPINS  ROT,144,T.DIM
   2328 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 60
DEFINITIONS     ERRORS

   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 61
DEFINITIONS     ERRORS

   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 62
DEFINITIONS     ERRORS

   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

```
APLO09  MACRO M1113  08-FEB-80 10:31  PAGE 63
DEFINITIONS     ERRORS

   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 64
DEFINITIONS     ERRORS

   3221 007400                          MOPINS  MMD,122
   3316 007400                                  LIST
```

```
APLOC9  MACRO M1113  08-FEB-80 10:31  PAGE 65
DEFINITIONS     ERRORS

   3318 007400                          DOPINS  DMD,122
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 66
DEFINITIONS     ERRORS

   3864 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 67
DEFINITIONS     ERRORS

   3866 007400                          DOPINS  MEM,70
   3921 007400                                  LIST
```

DEFINITIONS ERRORS

```
3923 007400                          MOPINS  NUB,106
3986 007400                                  LIST
3987
3988
3989
3990
3991
3992 007400                          DOPINS  UNON,106
4022 007400                                  LIST
4023
4024
4025
4026
4027
4028 007400                          DOPINS  INTR,66
4042 007400                                  LIST
4043
4044
4045
4046
4047
4048 007400                          DOPINS  SDIF,50
4161 007400                                  LIST
```

```
  4163 007400                            DOPINS  SCON,234
  4179 007400                                    LIST
  4180
  4181
  4182 007400                            DOPINS  CONT,112
  4200 007400                                    LIST
  4201
  4202
  4203 007400                            DOPINS  PSUB,236
  4209 007400                                    LIST
  4210
  4211
  4212 007400                            DOPINS  SUBS,116
  4288 007400                                    LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 70
DEFINITIONS     ERRORS

   4290 007400                          MOPINS  EXE,70
   4336 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 71
DEFINITIONS     ERRORS

   4338 007400                          MOPINS  ENCD,76
   4411 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 72
DEFINITIONS     ERRORS

   4413 007400                          MOPINS  FMT,160
   4423 007400                                  LIST
   4424
   4425
   4426
   4427
   4428
   4429 007400                          DOPINS  FMT,160
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 73
DEFINITIONS     ERRORS

   4953 007400                                  LIST
```

DEFINITIONS ERRORS

```
4955 007400                          MOPINS  IBM,132,,,MONIBM
4958 007400                                  LIST
4959
4960
4961 007400                          DOPINS  IBM,132
5046 007400                                  LIST
```

APL007

APL008

APL009

```
5048 007400                             MOPINS  QUAD,74,T.NIL+T.SPOP
5075 007400                                     LIST
5076
5077
5078
5079
5080
5081 007400                             DOPINS  QUAD,74,T.NIL+T.SPOP
5097 007400                                     LIST
5098
5099
5100
5101
5102
5103 007400                             MOPINS  QQAD,142,T.NIL+T.SFOP
5135 007400                                     LIST
5136
5137
5138
5139
5140
5141 007400                             DOPINS  QQAD,142,T.NIL+T.SPOP
5152 007400                                     LIST
5153
5154
5155
5156
5157
5158 007400                             MOPINS  DQAD,134,T.NIL+T.SPOP
5192 007400                                     LIST
5193
5194
5195
5196 007400                             DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 007400                                     LIST
```

```
5201 007400                          MOPINS  BXAV,162,T.NIL
5219 007400                                  LIST
5220
5221
5222
5223
5224
5225 007400                          MOPINS  BXCT,164,T.NIL+T.SPOP
5234 007400                                  LIST
523
5236
5237 007400                          DOPINS  BXCT,164,T.NIL+T.SPOP
5272 007400                                  LIST
5273
5274
5275
5276
5277
    007400                          MOPINS  BXIO,166,T.NIL+T.SPOP
528  007400                                  LIST
5288
5289
5290 007400                          DOPINS  BXIO,166 T.NIL+T.SPOP
5302 007400                                  LIST
5303
5304
5305
5306
5307 007400                          MOPINS  BXLC,212,T.NIL
5340 007400                                  LIST
5341
5342
5343
5344
5345
5346
5347 007400                          MOPINS  BXPP,170,T.NIL+T.SPOP
5356 007400                                  LIST
5357
5358
5359 00740C                          DOPINS  BXPP,170,T.NIL+T.SPOP
5381 007400                                  LIST
5382
5383
5384
5385
5386
5387
5388 007400                          MOPINS  BXPW,172,T.NIL+T.SPOP
5397 007400                                  LIST
5398
5399
5400 007400                          DOPINS  BXPW,172,T.NIL+T.SPOP
5417 007400                                  LIST
5418
5419
5420
```

```
5421
5422
5423 007400                          MOPINS  BXRL,174,T.NIL+T.SPOP
5432 007400                                  LIST
5433
5434
5435 007400                          DOPINS  BXRL,174,T.NIL+T.SPOP
5447 007400                                  LIST
5448
5449
5450
5451
5452
5453 007400                          MOPINS  BXTS,200,T.NIL
5562 007400                                  LIST
5563
5564
5565
5566
5567
5568
5569 007400                          MOPINS  BXWA,176,T.NIL
5592 007400                                  LIST
```

DEFINITIONS ERRORS

```
   5594 007400                          MOPINS  BXCR,206
   5716 007400                                  LIST
```

```
    5718 007400                          MOPINS  BXEX,202
    5775 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 79
DEFINITIONS     ERRORS

   5777 007400                          MOPINS  BXFX,210
   5879 007400                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 80
DEFINITIONS     ERRORS

   5881 007400                          MOPINS  BXNC,204
   5950 007400                                  LIST
```

DEFINITIONS ERRORS

```
 5952 007400                            MOPINS  BXNL,214
 5963 007400                                    LIST
 5964
 5965
 5966 007400                            DOPINS  BXNL,214
 6104 007400                                    LIST
```

DEFINITIONS ERRORS

```
 6106 007400                            DOPINS  FGET,150,T.DIM
 6330 007400                                    LIST
 6331
 6332
 6333
 6334
 6335
 6336 007400                            DOPINS  FPUT,152,T.DIM
 6485 007400                                    LIST
```

DEFINITIONS ERRORS

```
  6487 007400                          DOPINS  FPNT,154,T.DIM
  6714 00740C                                  LIST
```

```
6716 007400                             MOPINS  1SCN,130
6717                                    .SBTTL  OPERATORS               FIRST COORDINATE SCAN
6718
6719
6720 000000                                     CALL    RDSNC1          .Set up for first coordinate
6721 000004                                     CALLR   SCAN1           ;Do the scan
6723 000010                                     LIST
6724
6725
6726 000010                             MOPINS  SCAN,62,T.DIM
6727                                    .SBTTL  OPERATORS               SCAN
6728
6729 000010                             SCAN::  CALL    RDSNCK          ;Do error checks
6730 000014                                     CALLR   SCAN1           ;Do the scan
6732 000020                                     LIST
6733
6734
6735 000020                             MOPINS  1RDT,124
6736                                    .SBTTL  OPERATORS               FIRST COORDINATE REDUCTION
6737
6738 000020                                     CALL    RDSNC1          ;Setup for first coordinate reduction
6739 000024                                     CALLR   REDCT           ;Do the reduction
6741 000030                                     LIST
6742
6743
6744 000030                             MOPINS  RDCT,60,T.DIM
6745                                    .SBTTL  OPERATORS               REDUCTION
6746
6747 000030                             REDUCT::CALL    RDSNCK          ;Do error checks
6748 000034                                     CALLR   REDCT           ;And reduce
6749
6750
6751
6752                                    ;
6753                                    ;The SCAN Operator has been added in such a way to share a good part of
6754                                    ;Reduction's code.
6755                                    ;
6756                                    ; ADDED: May 1978               BY: Peter Haynes
6757                                    ;
6758                                    ;
6759                                    ; Use of temporary registers:
6760                                    ;
6761                                    ;       OPTMP0 - Pointer to dimension being reduced/scanned
6762                                    ;       OPTMP1 - Dimension temporary used by SATAN
6763                                    ;       OPTMP2 - Source pointer offset used by SATAN
6764                                    ;       OPTMP3 - Dimension to be Reduced/Scanned
6765                                    ;       OPTMP4 - Pointer to bottom of SP stack (high core)
6766                                    ;       OPTMP5 - Reduction/Scan flag
6767                                    ;       OPTMP6 - Result pointer offset/increment used by SATAN
6768                                    ;       OPTMP7 thru OPTM10 - Floating point identity for Reduction
6769                                    ;       OPTMP7  - Also used to hold result store routine for relational
6770                                    ;                 scans.
6771                                    ;
6772                                    ;This routine sets up for first coordinate operations
6773                                    ;
6774 000040  013704  003120'            RDSNC1: MOV     MONBLK,R4       ;Point to argument block
6775 000044  012702  000001                     MOV     #1,R2           ;Setup for reducing/scanning first dimension
```

```
6776 000050                                    CALL    FIXDIM          ;Get pointer to reduct/scan dimension
6777 000054  000137  000072'                   JMP     RDSNK1          ;Check for legal operation
6778                                           ;Note this disallows "f.CS scalar", to
6779                                           ; allow this form branch to RDSN2
6780                                   ;
6781                                   ;This routine gets the dimension argument if there is one and makes
6782                                   ;various error checks.
6783                                   ;
6784 000060                           RDSNCK: CALL    MFXDIM          ;Get pointer to dimension arg if any
6785 000064  005737  003112'                   TST     DIMBLK          ;Is there a dimension argument?
6786 000070  001404                            BEQ     RDSNK2          ;No, Ok
6787 000072  105764  000002           RDSNK1: TSTB    A.DIMS(R4)      ;Is the argument a scalar?
6788 000076  001001                            BNE     RDSNK2          ;No, Ok
6789 000100  104411                            RANERR                  ;Yes, RANK Error, A scalar has 0 dimensions
6790 000102  010237  005066'           RDSNK2: MOV     R2,OPTMP0       ;Save pointer to reduct/scan dimension
6791                                   ;
6792                                   ;Check for legality of Operation
6793                                   ;
6794 000106  016604  000010                    MOV     10(SP),R4       ;Get the operation number
6795 000112  016603  000006                    MOV     6(SP),R3        ;And its bits
6796 000116  032703  000001                    BIT     #T.OPFN,R3      ;Is it an operator?
6797 000122  001001                            BNE     1$              ;Yes, Ok
6798 000124  104407                            SYNERR                  ;No, SYNTAX Error
6799 000126  032703  000002           1$:     BIT     #T.SOP,R3       ;Is it a Primative Scalar Operator?
6800 000132  001001                            BNE     2$              ;Yes, Ok
6801 000134  104417                            DOMERR                  ;No, DOMAIN Error
6802 000136  026427  000000G 104407   2$:     CMP     DYDTAB(R4),#SYNERR  ;Is it a Monadic Scalar Operator?
6803 000144  001001                            BNE     3$              ;No, Ok
6804 000146  104417                            DOMERR                  ;Yes, DOMAIN Error
6805 000150  010437  005116'           3$:     MOV     R4,BLZBUB       ;Save the Operator Number
6806 000154                                    RETURN
6807                                   ;
6808                                   ;This routine handles the Reduction Operator
6809                                   ;
6810 000156  010237  005100'           REDCT:  MOV     R2,OPTMP5       ;Set Reduct/Scan flag for Reduction
6811 000162                                    CALL    RDSNGS          ;Get result space and build SP frames
6812 000166  005416                            NEG     (SP)            ;Make the offset for Satan negative
6813 000170  013766  003152'  000012          MOV     RESCVT,12(SP)   ;So copy case in SATAN works
6814 000176  013701  003140'                   MOV     RESPTR,R1       ;Point to result data
6815 000202  013703  005070'                   MOV     OPTMP1,R3       ;Get number of elements in Reduct dimension
6816 000206  001410                            BEQ     1$              ;None
6817 000210  005303                            DEC     R3              ;Cheat for Copy case
6818 000212  001406                            BEQ     1$              ;Still none
6819 000214                                    MUL     OPTMP2,R3       ;Compute the offset
6820 000230  010302                    1$:     MOV     R3,R2           ;Put in correct register
6821 000232  063702  003122'                   ADD     MONPTR,R2       ;Get pointer to source argument
6822 000236  013700  005074'                   MOV     OPTMP3,R0       ;Get number of dimensions to process
6823 000242  005300                            DEC     R0              ;The result will have one less dimensions
6824 000244  005037  005114'                   CLR     DEVIL           ;Don't increment the result pointer
6825 000250                                    CALL    SATAN           ;Recursive routine SATAN does reduction
6826 000254  010406                            MOV     R4,SP           ;Delete the SP frames
6827 000256                                    CALLR   RDSNRT          ;Clean up and return
6828                                   ;
6829                                   ; This routine handles the Scan operator
6830                                   ;
6831 000262  005037  005100'           SCAN1:  CLR     OPTMP5          ;Reset Reduct/Scan flag for Scan
6832 000266                                    CALL    RDSNGS          ;Get result space and build SP frames
```

```
6833 000272  013702  005116'                MOV     BLZBUB,R2          ;Get scalar function number
6834 000276  006202                         ASR     R2                 ;Convert function number to word pointer
6835 000300  132762  000200  000000G        BITB    #CS.ASS,DYDBTL(R2)  ;Is the function associative?
6836 000306  001017                         BNE     1$                 ;Yes, Then don't need extra SP frame
6837
6838                                                                    ;Non-associative scan, build final SP frame
6839 000310  013746  005104'                MOV     OPTMP7,-(SP)       ;Used only by relational scans
6840 000314  013746  005102'                MOV     OPTMP6,-(SP)       ;Result pointer increment
6841 000320  016646  000010                 MOV     10(SP),-(SP)       ;Stop count value
6842 000324  012746  000001                 MOV     #1,-(SP)           ;Initial count
6843 000330  011646                         MOV     (SP),-(SP)         ;Copy it for Count-limit increment
6844 000332  013746  005072'                MOV     OPTMP2,-(SP)       ;Offset for scan dimension
6845 000336  005416                         NEG     (SP)               ;Make it negative for non-associative functions
6846 000340  005037  005114'                CLR     DEVIL              ;Don't increment result pointer
6847 000344  000410                         BR      2$                 ;Let SATAN do the work
6848
6849                                                                    ;Associative scan
6850 000346  005337  005074'        1$:     DEC     OPTMP3             ;No extra frame to process
6851 000352  013737  005102'  005114'       MOV     OPTMP6,DEVIL       ;Result pointer increment, Assoc funcs are fast
6852 000360  013766  003152'  000012        MOV     RESCVT,12(SP)      ;So copy case of SATAN works
6853 000366  013701  003140'        2$:     MOV     RESPTR,R1          ;Point to result data
6854 000372  013702  003122'                MOV     MONPTR,R2          ;Point to starting place
6855 000376  013700  005074'                MOV     OPTMP3,R0          ;Get the number of frames to process
6856 000402  0052..  005074'                INC     OPTMP3             ;Fixup number of frames (origin one)
6857 000406                                 CALL    SATAN              ;Do the Scanning
6858 000412  010406                         MOV     R4,SP              ;Delete the SP frames
6859 000414                                 CALLR   RDSNRT             ;Clean up and return
6860                                ;
6861                                ;This routine sets up the argument fetch routine pointers, gets
6862                                ; space for the result and builds the SP stack frames for the
6863                                ; SATAN routine.
6864                                ;
6865
6866                                ;
6867                                ;Compute the size of the result data
6868                                ;
6869 000420  012701  000001    RDSNGS: MOV     #1,R1              ;Get the multiplicitive identity
6870 000424  013700  003...             MOV     MONBLK,R0          ;Get pointer to source argument
6871 000430  116002  000...             MOVB    A.DIMS(R0),R2      ;Get number of dimensions in source argument
6872 000434                             PUSH    R2                 ;Save it on the stack
6873 000436                             CONINC  R2                 ;Cheat for scalars
6874 000442  010237  005074'            MOV     R2,OPTMP3          ;Save it
6875 000446  005737  005100'            TST     OPTMP5             ;Are we Scanning?
6876 000452  001403                     BEQ     3$                 ;Yes
6877 000454  013716  005074'            MOV     OPTMP3,(SP)        ;Get the number of result dimensions
6878 000460  005316                     DEC     (SP)               ;Reduction result has one less dimensions
6879 000462  005716              3$:    TST     (SP)               ;Is the result a scalar?
6880 000464  001415                     BEQ     6$                 ;Yes, Then we're done
6881 000466  062700  000006            ADD     #A.DIM1,R0         ;Point to first dimension element
6882
6883 000472  020037  005100'    4$:     CMP     R0,OPTMP5          ;Is this the dimension ?
6884 000476  001...                     BEQ     5$                 ;Yes, Then skip it
6885 000500                             MUL     (R0),R1            ;No, Factor it into the result
6886 000512  005720              5$:    TST     (R0)+              ;Increment the pointer
6887 000514                             SOB     R2,4$              ;Loop for all dimensions
6888
6889 000520  010103              6$:    MOV     R1,R3              ;Save the number of elements (ALEN)
```

```
6890 000522 001002                      BNE     10$             ;The result is null
6891 000524 000137' 001020'             JMP     RDSN1           ;Go get space for the result
6892 000530 005737 005100'      10$:    TST     OPTMP5          ;Are we Scanning?
6893 000534 001403                      BEQ     1$              ;Yes, Skip identity code
6894 000536 005777 005100'              TST     @OPTMP5         ;Is the reduction dimension zero?
6895 000542 001516                      BEQ     2$              ;Yes, Do the identity code
6896                                    ;
6897                                    ;Compute the Result type and fetch/store routine pointers for non null arguments
6898                                    ;
6899 000544 006204              1$:     ASR     R4              ;Change function number into byte table pointer
6900 000546 116400 000000G              MOVB    DYDBTL(R4),R0   ;Get the function's bits
6901 000552 006304                      ASL     R4              ;Restore the function number
6902 000554 013702 003124'              MOV     MONTYP,R2       ;Get the source type
6903 000560 010237 003144'              MOV     R2,RESTYP       ;Assume copy case, Result type equals source
6904 000564 010201                      MOV     R2,R1           ;Copy the source type
6905 000566 006301                      ASL     R1              ;Multiply it by four
6906 000570 006301                      ASL     R1
6907 000572 060201                      ADD     R2,R1           ;Compute the fetch routine table offset
6908 000574 016137 000000G 003176'      MOV     LARGCT(R1),LARGCV ;Get the source fetch routine pointer
6909 000602 012737 104402 003220'       MOV     #SYSERR,RARGCV  ;Don't use the right fetch routine
6910 000610 016137 000000G 003152'      MOV     RESCT(R1),RESCVT ;Get the result store routine pointer
6911 000616 016237 000044' 003146'      MOV     SIZTAB(R2),RESSFT ;Get the result shift count
6912 000624 016237 000054' 003150'      MOV     LENTAB(R2),RESLEN ;Get the result element length
6913
6914 000632 027727 005066' 000001       CMP     @OPTMP0,#1      ;Is "the dimension" one?
6915 000640 003002                      BGT     26$             ;No, Not copy case
6916 000642 000137 001026'              JMP     RDSN2           ;Yes, Go get space for result
6917
6918 000646                     26$:    PUSH    R0              ;Save the function's bits
6919 000650                             CALL    CMPTRT          ;Calculate the result type
6920 000654 042726 177771               BIC     #^C<A.NUM>,(SP)+ ;Is the function a relation?
6921 000660 001030                      BNE     28$             ;No
6922 000662 005737 005100'              TST     OPTMP5          ;Yes, Are we scanning?
6923 000666 001034                      BNE     30$             ;No
6924 000670 013700 003124'              MOV     MONTYP,R0       ;Get the source type
6925 000674 001001                      BNE     27$             ;Is the source type character?
6926 000676 104417                      DOMERR                  ;Yes, Can't mix character and Boolean types
6927                                                            ;Relational scans
6928 000700 060002              27$:    ADD     R0,R2               ;Compute offset to fetch routine tables
6929 000702 016037 000000G 005104'      MOV     BRESCT(R0),OPTMP7   ;For relational scans
6930 000710 016037 000044' 003146'      MOV     SIZTAB(R0),RESSFT   ;Set the result shift count
6931 000716 016037 000054' 003150'      MOV     LENTAB(R0),RESLEN   ;Set the result element length
6932 000724 016237 000000G 003220'      MOV     RARGCT(R2),RARGCV   ;Set the right argument fetch routine pointer
6933 000732 016237 000000G 003152'      MOV     RESCT(R2),RESCVT    ;Set the initial result store routine pointer
6934 000740 000427                      BR      RDSN1               ;Go get some space
6935                                                            ;Normal case
6936 000742 063700 003124'      28$:    ADD     MONTYP,R0           ;Compute the source fetch routine pointer
6937 000746 016037 000000G 003176'      MOV     LARGCT(R0),LARGCV   ;Get the source fetch routine pointer
6938 000754 010437 005116'              MOV     R4,BLZBUB           ;Save the function number, it may have changed
6939 000760 060201              30$:    ADD     R2,R1               ;Compute the right fetch routine offset
6940 000762 016137 000000G 003220'      MOV     RARGCT(R1),RARGCV   ;Get the right fetch routine pointer
6941 000770 013737 003152' 005104'      MOV     RESCVT,OPTMP7       ;So the copy case in SATAN works
6942 000776 000413                      BR      RDSN2               ;Go get some space for the result
6943                                    ;
6944                                    ;The reduction dimension is zero, Return an array of identity elements
6945                                    ;
6946 001000 005003              2$:     CLR     R3              ;There is no second argument
```

```
6947 001002  013702  003120'                  MOV     MONBLK,R2       ;Get pointer to argument
6948 001006                                   CALL    GETIDN          ;Get an array of identity elements
6949 001012  005726                           TST     (SP)+           ;Clean result rank off stack
6950 001014                                   CALLR   RDSNRT          ;All done, go clean up
6951
6952                                  ;
6953                                  ;Calculate the amount of space needed and build the result block
6954                                  ;
6955 001020  013737  003124' 003144' RDSN1:  MOV     MONTYP,RESTYP   ;Nothing special, Result type is same as source
6956 001026  013704  005074'         RDSN2:  MOV     OPTMP3,R4       ;Get the argument's rank
6957 001032  013702  003120'                 MOV     MONBLK,R2       ;Get pointer to the argument
6958 001036                                  CALL    GETSPA          ;Get space for the result and fill in the header
6959 001042  013701  003120'                 MOV     MONBLK,R1       ;Get pointer to the argument
6960 001046  005703                          TST     R3              ;Anything there?
6961 001050  001002                          BNE     17$             ;Yes, Create SP frames
6962 001052                                  CALLR   RDSNRT          ;No, then return
6963                                  ;
6964                                  ;Create special frames on the SP stack for the SATAN routine
6965                                  ;
6966 001056  010637  005076'         17$:    MOV     SP,OPTMP4       ;Save the Stack pointer
6967 001062  013737  005074' 005114'         MOV     OPTMP3,DEVIL    ;Copy the number of source dimensions
6968 001070  016102  000004                  MOV     A.ALEN(R1),R2   ;Get the source size
6969 001074  062701  000006                  ADD     #A.DIM1,R1      ;Point to source Dimension List
6970 001100  013700  003136'                 MOV     RESULT,R0       ;Point to result block
6971 001104  016004  000004                  MOV     A.ALEN(R0),R4   ;Get the result size
6972 001110  062700  000006                  ADD     #A.DIM1,R0      ;Point to result dimension list
6973
6974 001114  010203                  18$:    MOV     R2,R3           ;Copy current length
6975 001116  005002                          CLR     R2              ;Clear for Divide
6976 001120  020137  005066'                 CMP     R1,OPTMP0       ;Is this the dimension?
6977 001124  001050                          BNE     21$             ;No
6978 001126  005711                          TST     (R1)            ;Is it zero?
6979 001130  001405                          BEQ     19$             ;Yes, don't Divide
6980 001132                                  DIV     (R1),R2         ;Divide By Size of this dimension
6981 001144  010203                  19$:    MOV     R2,R3           ;Copy the source dimension offset
6982 001146                                  ASH     MONSFT,R3       ;Convert the source offset to bytes
6983 001162  010337  005072'                 MOV     R3,OPTMP2       ;Save dimension offset (if dim=0, so is R2)
6984 001166  012137  005070'                 MOV     (R1)+,OPTMP1    ;Save length of dimension
6985 001172  005737  005100'                 TST     OPTMP5          ;Are we Scanning?
6986 001176  001070                          BNE     23$             ;No
6987 001200                                  PUSH    R2              ;Save the source offset
6988 001202  005002                          CLR     R2              ;Setup for divide
6989 001204  010403                          MOV     R4,R3           ;Get the result pointer offset
6990 001206                                  DIV     (R0)+,R2        ;Divide by the result dimension
6991 001220  010204                          MOV     R2,R4           ;Save the new result pointer offset
6992 001222                                  ASH     RESSFT,R2       ;Convert the result pointer offset to bytes
6993 001226  010237  005102'                 MOV     R2,OPTMP6       ;Save it
6994 001242                                  POP     R2              ;Restore the source offset
6995 001244  000445                          BR      23$             ;Go see if we're done
6996
6997 001246                          21$:    DIV     (R1),R2         ;Divide by Dimension Size
6998 001260                                  PUSH    R2              ;Save the source offset
6999 001262  005002                          CLR     R2              ;Setup for divide
7000 001264  010403                          MOV     R4,R3           ;Copy result offset
7001 001266  005711                          TST     (R0)            ;Dimension zero?
7002 001270  001405                          BEQ     22$             ;Yes, Then don't divide
7003 001272                                  DIV     (R0)+,R2        ;Divide by next dimension size
```

```
7004 001304  010204                 22$:  MOV     R2,R4           ;Copy the result offset
7005 001306                               POP     R2              ;Restore the source offset
7006 001310  005046                       CLR     -(SP)           ;Clear result pointer
7007 001312  010403                       MOV     R4,R3           ;Copy the offset
7008 001314                               ASH     RESSFT,R3       ;Convert the result offset to bytes
7009 001330  010346                       MOV     R3,-(SP)        ;Store the result offset
7010 001332  005046                       CLR     -(SP)           ;Clear source pointer
7011 001334  012146                       MOV     (R1)+,-(SP)     ;Store Count-limit
7012 001336  005046                       CLR     -(SP)           ;Clear Count
7013 001340  010203                       MOV     R2,R3           ;Copy the offset
7014 001342                               ASH     MONSFT,R3       ;Convert the source offset to bytes
7015 001356  010346                       MOV     R3,-(SP)        ;Store source offset for this dimension
7016 001360  005337  005114'        23$:  DEC     DEVIL           ;Used all the dimensions yet?
7017 001364  001253                       BNE     18$             ;No, Loop
7018 001366  005046                       CLR     -(SP)           ;Clear result pointer
7019 001370  005737  005100'             TST     OPTMP5          ;Are we scanning?
7020 001374  001003                       BNE     24$             ;No
7021 001376  013746  005102'             MOV     OPTMP6,-(SP)    ;Yes, Fill in the result pointer increment
7022 001402  000401                       BR      25$
7023 001404  005046                 24$:  CLR     -(SP)           ;Clear the result pointer increment
7024 001406  005046                 25$:  CLR     -(SP)           ;Clear source pointer
7025 001410  013746  005070'             MOV     OPTMP1,-(SP)    ;Store Count-limit
7026 001414                               CONINC  <(SP)>          ;Make sure SATAN has something to count
7027 001420  005046                       CLR     -(SP)           ;Clear Count
7028 001422  013746  005072'             MOV     OPTMP2,-(SP)    ;Store source offset for this dimension
7029 001426                               PUSH    <@OPTMP4>       ;Get return address from under SP frames
7030 001432  013704  005076'             MOV     OPTMP4,R4       ;Get pointer top frame
7031 001436                               RETURN
7032
7033                                ;
7034                                ;This routine cleans up after completing the reduction/scan and returns
7035                                ;
7036
7037 001440  005726                RDSNRT: TST     (SP)+           ;Get rid of old return address (from RDSNGS)
7038 001442                               POP     R1              ;Save the Return address
7039 001444                               POP     <R2,R3,R4>      ;Restore the registers
7040 001452                               CALL    GETNXT          ;Get the Next Token
7041 001456                               PUSH    <R4,R3,R2,R1>   ;Put the arguments back
7042 001466                               RETURN                  ;And Return
7044 001470                               LIST
```

```
   7046 001470                          MOPINS  GETIDN,0
   7193 001470                                  LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 86
OPERATORS                REDUCTION

   7195 001470                              MOPINS  SATAN,0
   7404 001470                                      LIST
```

```
APL009  MACRO M1113  08-FEB-80 10:31  PAGE 87
OPERATORS                REDUCTION

   7406 001470                          MOPINS  OUTR,156
   7476 001470                                  LIST
```

```
 7478 001470                            MOPINS  DOUTER,0
 7504 001470                                    LIST
 7505
 7506
 7507
 7508
 7509
 7510 001470                            MOPINS  PRDSET,0
 7564 001470                                    LIST
 7565
 7566
 7567
 7568
 7569
 7570 001470                            MOPINS  INSPAC,0
 7612 001470                                    LIST
```

```
   7614 001470                          DOPINS  INNR,156
```

```
   7960 001470                                    LIST
```

```
   7962 001470                          MOPINS  SUBR,0
```

```
   8725 001470                                    LIST
```

```
   8727 001470                          MOPINS  RAND,0
   8771 001470                                  LIST
```

```
OPERATORS                REDUCTION

      1          000001                              .END
```

APL009 MACRO M1113 08-FEB-80 10:31 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    C.ALOC= 000162    D.CONT= 000006    EDICNT  005064R  025
AC1    =%000001          CHRLBR= 000073    C.BLOK= 000064    D.CTLM= 000000    EDIFLG  005062R  025
AC2    =%000002          CHRLPR= 000072    C.BLSZ= 000140    D.DEAL= 000003    EDIT   = 000003
AC3    =%000003          CHRMNS= 000137    C.BUFF= 177760    D.DECD= 000002    EIS    = 000000
AC4    =%000004          CHRMUL= 000075    C.BUFL= 177762    D.DIV = 000000    ELESIZ  005132R  025
AC5    =%000005          CHRNEG= 000100    C.CCNT= 177766    D.DMD = 000010    ENDCOR  000004R  025
ALLINT  003274R     025  CHRPDL= 000271    C.CPTR= 177764    D.DQAD= 177777    EOF    = 104475
ARYRET  003276R     025  CHRPLU= 000055    C.CZ  = 001000    D.DROP= 000002    ERRMAX= 000166
ASCQUO= 000042           CHRPWR= 000120    C.DIG = 000040    D.ENCD= 000002    ERRNUM  004772R  025
A.ALEN= 000004           CHRQST= 000121    C.FASW= 020000    D.EQ  = 000000    ERRPTR  004770R  025
A.ARG = 000040           CHRQUO= 000113    C.FLEN= 000010    D.EXPD= 000002    ERRO    004774R  025
A.ASB = 000200           CHRRBR= 000047    C.FLSH= 004000    D.FGET= 000005    EXEBUF  005054R  025
A.BITS= 000003           CHRRPR= 000042    C.FRC = 000001    D.FMT = 000004    E$CT    005056R  025
A.CON = 000020           CHRSEM= 000074    C.IDEN= 000010    D.FPNT= 000005    E$PT    005060R  025
A.DIMS= 000002           CHRSMO= 000112    C.NAME= 000102    D.FPUT= 000005    FA.APD= 000100
A.DIM1= 000006           CHRSPA= 000040    C.NOBF= 000400    D.GE  = 000000    FA.CRE= 000010
A.NUM = 000006           CHRUPA= 000131    C.NUM = 000020    D.GT  = 000000    FA.DLK= 001000
A.PTR = 000000           CHR.BS= 000010    C.OP  = 000004    D.IBM = 000002    FA.ENB= 100000
A.TMP = 000010           CHR.CC= 000003    C.OPN = 010000    D.IND = 000002    FA.EXC= 002000
BADDIR= 104463           CHR.CR= 000015    C.RDLN= 177776    D.INNR= 000013    FA.EXT= 000004
BADERR= 104437           CHR.CZ= 000032    C.REWT= 002000    D.INTR= 000006    FA.NSP= 000100
BL2BUB  005116R     025  CHR.GT= 000046    C.RND = 000010    D.LE  = 000000    FA.NSR= 000200
BRESCT= ****** GX        CHR.LF= 000012    C.SBIT= 177756    D.LOG = 000001    FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043    C.SPFX= 000100    D.LT  = 000000    FA.RD = 000001
CCNT    005072R     025  CHR.SI= 000017    C.SPGL= 000200    D.MAX = 000000    FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057    C.SPLX= 000001    D.MEM = 000002    FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016    C.SPRS= 000002    D.MIN = 000000    FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011    C.SRC = 177774    D.MUL = 000000    FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106    C.SRCP= 000024    D.NE  = 000000    FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141    C.TT  = 000004    D.NN  = 000000    FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDC= 177770    D.NR  = 000000    FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163    C.WIDR= 177772    D.OR  = 000000    FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164    DEBUG = 000000    D.PLUS= 000000    FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165    DEFERR= 104405    D.PSUB= 000006    FD.CR = 000002
CHAN08  001766R     025  CHR..Z= 000172    DEVIL   005114R  025 D.PWR = 000001    FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060    DEVNTF= 104470    D.QQAD= 000000    FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R  025 D.QUAD= 000000    FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001    DIMAX = 000020    D.RES = 000000    FD.INS= 000010
CHAN12  002676R     025  CMDBUF  003250R  025 DIMBLK  003112R  025 D.RHO = 000000    FD.ISP= 002000
CHAN13  000022R     023  CMDMAX= 000010    DIMPTR  003114R  025 D.ROT = 000003    FD.MNT= 100000
CHAN14  000204R     023  CMPTRT= ****** GX DIMTYP  003116R  025 D.SCON= 000006    FD.OSP= 004000
CHNOPN= 104471           CNTSIZ  005130R  025 DIRFUL= 104466    D.SDIF= 000006    FD.PLC= 000004
CHRAND= 000051           COMBIN  005076R  025 DOMERR= 104417    D.SUB = 000000    FD.PRN= 000004
CHRBEL= 000007           COMCNT  005126R  025 DPTERR= 104416    D.SUBS= 000006    FD.PSE= 010000
CHRBKR= 000133           COMERR= 104425    DYDBTL= ****** GX    D.TAKE= 000002    FD.RAH= 000001
CHRBOL= 000200           COMPTR  005122R  025 DYDCNT  003224R  025 D.TRAN= 000004    FD.RAN= 000002
CHRBOX= 000114           COMRTN  003156R  025 DYDTAB= ****** GX D.UNON= 000006    FD.REC= 000001
CHRCMA= 000054           COMSKP  005124R  025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRCOL= 000076           CORPTR  005120R  025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRDEL= 000107           CS.ASS= 000200    D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRDIV= 000053           CS.BOO= 000010    D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDLR= 000176           CS.FLO= 000020    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDLT= 000110           CS.IDN= 000040    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDOT= 000056           CS.ONE= 000100    D.BXRL= 000007    D.1COM= 000001    FF.NV  = 000003
CHREP  = 000105          CTEPS   000152R  025 D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRERR= 104431           CTMAX   000212R  025 D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRFIL= 000177           CTRMAX  000172R  025 D.COMP= 000001    EDICHR  005063R  025 FF.RWF= 000006
```

APLOO9 MACRO M1113 08-FEB-80 10:31 PAGE 94-2
SYMBOL TABLE

```
FF.SPC= 000004          F.FFBY= 000014          GETSPA= ****** GX       L.CNT = 000002          M.NEG = 000000
FHEADR  005012R     025 F.FLAG= 000002          GNOME   003102R     025 L.PRNT= 000001          M.NOT = 000000
FHRET   005040R     025 F.FLB = 000100          HASHC   005136R     025 L.PTR = 000000          M.NUB = 000006
FILNTF= 104467          F.FNAM= 000110          HASHTB  005140R     025 L.PTR1= 000004          M.NUM = 000204
FIS   = 000000          F.FNB = 000102          HSHMSK= 177601          L.SUBS= 000002          M.OUTR= 000012
FISCON= 046000          F.FNDF= 000010          HSHSIZ= 000200          MACHIN= 000106          M.PITM= 000000
FIXDIM= ****** GX       F.FTYP= 000116          IASH  = ****** GX       MAXCHN= 000014          M.PRDS= 000006
FIXPTR  003074R     025 F.FUN = 000010          ICHAN   005050R     025 MFXDIM= ****** GX       M.QQAD= 000000
FLAGS   004764R     025 F.FVER= 000120          IDIV  = ****** GX       MONBLK  003120R     025 M.QUAD= 000000
FLCBUF  003250R     025 F.HEAD= 000006          ILFNAM= 104464          MONCVT  003132R     025 M.RAND= 000012
FLINE   005014R     025 F.HIBK= 000004          ILLCHN= 104540          MONLEN  003130R     025 M.RAVL= 000000
FLOEXP  003270R     025 F.LBL = 000020          ILLSW = 104565          MONPTR  003122R     025 M.RCIP= 000000
FLONEG  003272R     025 F.LINE= 000012          IMUL  = ****** GX       MONSFT  003126R     025 M.RDCT= 000011
FLSWTB  000222R     025 F.LOC = 000004          INDERR= 104410          MONTYP  003124R     025 M.REVS= 000003
FLTLEN= 000010          F.LST = 000004          INNTMP  003250R     025 MONVAL  003134R     025 M.ROLL= 000000
FNINSI  005044R     025 F.LUN = 000042          INNTM2  005016R     025 M$RDCT  000030RG    020 M.SATA= 000001
FORIGN  000032R     025 F.MACT= 000054          INPTR   000064R     025 M$SCAN  000010RG    020 M.SCAN= 000011
FO.APD= 000106          F.MBC1= 000055          INSNAM= 104462          M$1RDT  000020RG    020 M.SHAP= 000000
FO.MFY= 000002          F.MBFG= 000056          IOERR = 104477          M$1SCN  000000RG    020 M.SIGN= 000000
FO.RD = 000001          F.MON = 000002          I.CNT = 177776          M.ABS = 000000          M.SUBR= 000001
FO.UPD= 000006          F.NIL = 000001          I.DIM = 177772          M.BOL = 000200          M.SYM = 000203
FO.WRT= 000016          F.NRBD= 000024          I.LEN = 000012          M.BXAV= 000007          M.SYN = 000201
FPMODE  000042R     025 F.NREC= 000030          I.OERR= 000062          M.BXCR= 000007          M.TRAN= 000012
FPP   = 000000          F.OPN = 000010          I.OLD = 177770          M.BXCT= 000007          M.$ABS= 000000
FPPLEN= 000004          F.OVBS= 000030          I.PTR = 177774          M.BXEX= 000007          M.$CEI= 000000
FREBLK  007400R     025 F.PREV= 000000          I.STR = 177766          M.BXFX= 000007          M.$FLR= 000000
FUZZ    000142R     025 F.PRNT= 100000          LABERR= 104406          M.BXIO= 000007          M.$IDN= 000000
FXPTR1  003076R     025 F.PROT= 010000          LABLPT  005042R     025 M.BXLC= 000003          M.$NEG= 000000
F$EVAL= 000100          F.RACC= 000016          LARGBL  003162R     025 M.BXNC= 000007          M.$SIG= 000000
F$EVI = 000200          F.RATT= 000001          LARGCT= ****** GX       M.BXNL= 000007          M.1RDT= 000011
F$EXEC= 000400          F.RCNM= 000034          LARGCV  003176R     025 M.BXPP= 000007          M.1REV= 000003
F$FUNC= 001000          F.RCTL= 000017          LARGFL  003160R     025 M.BXPW= 000007          M.1SCN= 000011
F.ACTL= 000076          F.READ= 004000          LARGLE  003174R     025 M.BXRL= 000007          NAMBUF  003240R     025
F.ALOC= 000040          F.REDF= 000040          LARGLN  003166R     025 M.BXTS= 000007          NB.DEV= 000200
F.BBFS= 000062          F.RES = 000040          LARGPT  003164R     025 M.BXWA= 000007          NB.DIR= 000100
F.BDB = 000070          F.RET = 177772          LARGSF  003172R     025 M.CEI = 177777          NB.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002          LARGTY  003170R     025 M.CHR = 000202          NB.SD1= 000400
F.BITS= 177774          F.RTYP= 000000          LARGVA  003200R     025 M.CMT = 000206          NB.SD2= 001000
F.BKDN= 000026          F.R2  = 000014          LBINC   005100R     025 M.DOUT= 000006          NB.SNM= 000040
F.BKDS= 000020          F.R3  = 000016          LBUFF   003316R     025 M.DQAD= 000006          NB.STP= 000020
F.BKEF= 000050          F.R4  = 000020          LBUFLN= 000621          M.ENCD= 000004          NB.SVR= 000010
F.BKP1= 000051          F.SEQN= 000100          LLOUNT  005106R     025 M.EXE = 000000          NB.TYP= 000002
F.BKST= 000024          F.SPDV= 000072          LENERR= 104412          M.EXP = 000001          NB.VER= 000001
F.BKVB= 000064          F.SPUN= 000074          LENTAB  000054R     025 M.FACT= 000001          ND.DFV= 000200
F.BR  = 000020          F.STBK= 000036          LEXPTR  003100R     025 M.FIL = 000177          ND.EXT= 000002
F.CHR = 000075          F.STK = 000022          LFTOP   005066R     025 M.FLR = 177777          ND.NAM= 000004
F.CNTG= 000034          F.STOP= 000004          LINC    005102R     025 M.FMT = 000004          ND.PPN= 000100
F.CODE= 000010          F.STP = 000001          LINE    005016R     025 M.GDUP= 000003          ND.SEX= 000000
F.DFNB= 000046          F.SUSP= 000002          LININC  005026R     025 M.GDWN= 000003          ND.SES= 000020
F.DSPT= 000044          F.TRC = 000002          LISIZE= 000020          M.GETI= 000006          ND.SNM= 000000
F.DVNM= 000134          F.UNIT= 000136          LITOP   005072R     025 M.IBM = 000003          ND.SNS= 000040
F.EFBK= 000010          F.URBD= 000020          LNLNG = 104541          M.IDN = 000000          ND.SPN= 001400
F.EFN = 000050          F.VAL = 177776          LOCLCT  005036R     025 M.INSP= 000006          NEGSGN  005063R     025
F.END = 002000          F.VBN = 000064          LOWCOR  000002R     025 M.IOTA= 000000          NINPT1  003302R     025
F.EOPB= 000032          F.VBSZ= 000060          LSTERP  004766R     025 M.LIT = 000205          NINPT2  003300R     025
F.ERR = 000052          GETIDN= ****** GX       LWIDTH  000070R     025 M.LN  = 000001          NINSTK  003304R     025
F.FACC= 000043          GETNXT= ****** GX       L.BITS= 000003          M.MMD = 000010          NINTMP  003310R     025
```

SYMBOL TABLE

```
NINTYP  003306R     025 OPTM14  005026R     025 RINC    005104R     025 SYMPT1  005346R     025 TYPE  = 000166
NOTAVL= 104472          ORIGIN  000030R     025 RLTBAS  005066R     025 SYMTAB  005352R     025 T.BOL = 020000
NOTOPN= 104473          OUTPTR  000066R     025 ROTINC  005070R     025 SYMTBF= 104404          T.CON = 004000
NPRT1   004762R     025 OVROPR= 000011          ROUND   000010R     025 SYMTOP  005350R     025 T.DIM = 000100
NULL    007364R     025 PATCH   007344R     025 ROVER   003312R     025 SYMTP   006272R     025 T.FUN = 010000
NUMCHN= 000020          PRECSN  000052R     025 RSTS  = 000000          SYMURN= 000054          T.LIST= 000400
NUMDFL  003230R     025 RANDAC  003064R     025 RSX   = 000115          SYNERR= 104407          T.LIT = 000020
NUMEFL  003233R     025 RANDY   003062R     025 RTOP    005070R     025 SYSERR= 104402          T.MON = 000010
NUMESN  003236R     025 RANERR= 104411          RT11  = 000000          SYSTAT  000000R     024 T.NIL = 000004
NUMFLG  003230R     025 RARGBL  003204R     025 R.BITS= 000003          SY.BAD= 000001          T.NUM = 002000
NUMLEN  000062R     025 RARGCT= ****** GX       R.FIX = 000001          SY.BRK= 100000          T.OPFN= 000001
NUMNEG  003232R     025 RARGCV  003220R     025 R.LCT = 000002          SY.BS = 000010          T.OTHR= 040000
NUMQFL  003234R     025 RARGFL  003202R     025 R.LN  = 000002          SY.CLR= 000040          T.SOP = 000002
NUMRIG  003231R     025 RARGLE  003216R     025 R.LNUM= 000010          SY.ERP= 000020          T.SPOP= 000040
NUMWID  003226R     025 RARGLN  003210R     025 R.NEXT= 000006          SY.NOU= 000100          T.TMP = 000200
NUMZFL  003235R     025 RARGPT  003206R     025 R.OFF = 000000          SY.OPN= 000002          T.VAL = 100000
NYIERR= 104432          RARGSF  003214R     025 R.PREV= 000004          SY.PMT= 001000          T.VAR = 001000
N.BLSZ= 000036          RARGTY  003212R     025 R.PTR = 000000          SY.SRT= 000400          UPPER = 177400
N.DESC= 000020          RARGVA  003222R     025 R.SEQ = 000003          SY.SVC= 000004          UTNAME  000344R     023
N.DEV = 000032          RBASE   005074R     025 R.VAR = 000002          SY.SVL= 000200          VALERR= 104413
N.DEVN= 000034          RCOUNR  005112R     025 R6    =%000006          S.AB  = 177776          VARBLE  007400R     025
N.DID = 000024          RCOUNT  005110R     025 R7    =%000007          S.ASB = 100000          VARBLS  000000R     025
N.DVNM= 000032          RDSNCK  000060R     020 SATAN = ****** GX       S.BFHD= 000020          VERNUM  000000R     025
N.EXT = 000014          RDSNC1  000040R     020 SCAN    000010RG    020 S.BITS= 000002          VRSION= 000002
N.FID = 000000          RDSNGS  000420R     020 SCAN1   000262R     020 S.CHR1= 000003          WIDTH   000072R     025
N.FNAM= 000006          RDSNK1  000072R     020 SI    =%000005          S.FATT= 000016          WROUND  000020R     025
N.FTYP= 000014          RDSNK2  000102R     020 SIDAMN= 104433          S.FDB = 000140          WSDATE  000136R     025
N.FVER= 000016          RDSNRT  001440R     020 SIFRAM  003314R     025 S.FLB = 040000          WSFULL= 104403
N.NAME= 000006          RDSN1   001020R     020 SISTAK  006350R     025 S.FNAM= 000006          WSNAME  000074R     025
N.NEXT= 000022          RDSN2   001026R     020 SITOP   006346R     025 S.FNB = 000036          WSTIME  000132R     025
N.PPN = 000024          REDCNT  005070R     025 SITP    007144R     025 S.FNBW= 000017          $FIS  = ****** G
N.STAT= 000020          REDCT   000156R     020 SIWARN= 000200          S.FNTY= 000004          $FPP  = ****** G
N.UNIT= 000034          REDIGN  005066R     025 SIZTAB  000044R     025 S.FTYP= 000002          $FPPLN= ****** G
OBUFLN= 001000          REDOFF  005072R     025 SI.BIT= 177774          S.LEN = 000002          $MACHN= ****** G
OCHAN   005052R     025 REDUCT  000030RG    020 SI.LEN= 177772          S.LINK= 177774          $RSTS = ****** G
OMODE   000402R     023 REENTR  003060R     025 SI.RET= 177772          S.NFEN= 000020          $RSX  = ****** G
OPTMP0  005066R     025 REGTYP= 000003          SI.VAL= 177776          S.TYPE= 140000          $RT11 = ****** G
OPTMP1  005070R     025 RESBIT  003142R     025 SKPSIZ  005134R     025 S.VAL = 000000          .BOOLE= 000002
OPTMP2  005072R     025 RESCT = ****** GX       SPHOLD  000004R     024 TKDIMS  005072R     025 .CHARA= 000000
OPTMP3  005074R     025 RESCVT  003152R     025 SPINIT  000002R     024 TKFRAM  005066R     025 .DEF. = 000002
OPTMP4  005076R     025 RESERR= 104440          STKINC= 000774          TKLEFT  005070R     025 .FLOAT= 000006
OPTMP5  005100R     025 RESLEN  003150R     025 STOP  = 104400          TMODE   000403R     023 .FSRCB= ****** G
OPTMP6  005102R     025 RESPTR  003140R     025 SUBCNT  003104R     025 TMPTAG= 000166          .INTEG= 000004
OPTMP7  005104R     025 RESSFT  003146R     025 SUBDCT  003106R     025 TTYBUF  004140R     025 .MOLUN  000404R     023
OPTMP8  005106R     025 RESTYP  003144R     025 SUBPTR  003110R     025 TTYLEN= 000622          ...GBL= 000000
OPTMP9  005110R     025 RESULT  003136R     025 SYMB1   005344R     025 TURKEY= 104436          ...TPC= 000020
OPTM10  005112R     025 RESVAL  003154R     025 SYMF    005340R     025
```

```
. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
```

```
APLO09  MACRO M1113  08-FEB-80 10:31  PAGE 94-4
SYMBOL TABLE

CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    001470      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13391 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:52
APLO09,DM1:[305,301]APLO09/-SP=APLRX1,APLDEF,APLO09,APLOP,END
```

FILEID**APL010

```
  AAAAAA      PPPPPPPP      LL              000000        11        000000
  AAAAAA      PPPPPPPP      LL              000000        11        000000
AA      AA    PP      PP    LL            00      00    1111      00      00
AA      AA    PP      PP    LL            00      00    1111      00      00
AA      AA    PP      PP    LL            00      00      11      00    0000
AA      AA    PP      PP    LL            00      00      11      00    0000
AA      AA    PPPPPPPP      LL            00      00      11      00  00  00
AA      AA    PPPPPPPP      LL            00      00      11      00  00  00
AAAAAAAAAA    PP            LL            00      00      11      0000    00
AAAAAAAAAA    PP            LL            00      00      11      0000    00
AA      AA    PP            LL            00      00      11      00      00      ....
AA      AA    PP            LL            00      00      11      00      00      ....
AA      AA    PP            LLLLLLLLLL      000000      111111      000000        ....
AA      AA    PP            LLLLLLLLLL      000000      111111      000000        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 10:35
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS               RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    59- 2331    MIXED FUNCTION            MONADIC TRANSPOSE
```

APL010 MACRO M1113 08-FEB-80 10:35
TABLE OF CONTENTS

87- 7407 OPERATORS OUTER PRODUCT
93- 8728 SUBROUTINES .RAND RANDOM NUMBER GENERATOR

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 1


     1          000000                RT11    =  0          ;NOT RT-11
     2          000000                RSTS    =  0          ;NOT RSTS/E
     3          000115                RSX     =  'M         ;RSX-11/M
     4
     5          000000                EIS     =  0          ;NO EIS
     6          000000                FIS     =  0          ;NO FIS
     7          000000                FPP     =  0          ;NO FPP
     8
     9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
    10
    11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APLO10
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLO10 IS:
      9                                 ;
     10                                 ;       APLO10 < ...,APLO10,APLOP
     11                                 ;
     12
     13          000012                OVROPR= 10.                     ;OVERLAY SPECIFIED OPERATORS
```

```
      5
      6                                 ;++
      7                                 ; FUNCTIONAL DESCRIPTION:
      8                                 ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
      9                                 ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
     10                                 ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
     11                                 ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
     12                                 ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
     13                                 ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
     14                                 ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
     15                                 ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
     16                                 ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
     17                                 ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
     18                                 ;       INTACT!
     19                                 ;--
     24
     27
    124
    125                                 ;THESE MACROS ARE FOR OVFRLAY CODE.   THEY ARE USED THE SAME AS
    126                                 ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
    127                                 ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
    128                                 ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
    129                                 ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
    130                                 ;MACROS.
    131
    132                                 .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
    133                                 .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
    134                                 .IF EQ OVROPR-D.'NAME
    135                                         .PSECT  CODE
    136                                 D$'NAME::
    137                                 .ENDM
    138
    139                                 .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
    140                                 .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
    141                                 .IF EQ OVROPR-M.'NAME
    142                                         .PSECT  CODE
    143                                         .IF NE NUM
    144                                 M$'NAME::
    145                                     .ENDC
    146                                 .ENDM
    147
    148                                 .MACRO  LIST
    149                                         .ENDC
    150                                 .ENDM
    151
    152                                 .MACRO  PRMRES
    153                                         .IF EQ 1
    154                                 .ENDM
    155
    159                                         ;*********** ASSEMBLING THE FILE!!!!!
    160 007400                          PRMRES
    199 007400                                  LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,.INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,.INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,.INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,.FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,.FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,.FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,.INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,.INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 47-1
DEFINITIONS     EPRORS

     618
     619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
     639 007400                                  LIST
```

```
641 007400                          DOPINS  AND,42,T.SOP,,BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,,BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,,BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,,BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,,CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,,CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,,CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.SOP,,CHARACTER+CS.IDN
780 007400                                  LIST
781
```

```
    783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    786 007400                                  LIST
    787
    788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    791 007400                                  LIST
    792
    793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
    799 007400                                  LIST
    800
    801
    802 007400                          MOPINS  $IDN,-2
    808 007400                                  LIST
    809
    810
    811 007400                          MOPINS  $FLR,-14
    855 007400                                  LIST
    856
    857
    858 007400                          MOPINS  $CEI,-16
    868 007400                                  LIST
    869
    870
    871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
    878 007400                                  LIST
    879
    880 007400                          MOPINS  $NEG,-4
    887 007400                                  LIST
    888
    889
    890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
    901 007400                                  LIST
    902
    903 007400                          MOPINS  $SIG,-6
    914 007400                                  LIST
    915
    916
    917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
    928 007400                                  LIST
    929
    930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
    939 007400                                  LIST
    940
    941
    942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
    949 007400                                  LIST
    950
    951
    952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
    959 007400                                  LIST
    960
    961 007400                          PRMRES
    966 007400                                  LIST
    967
    968
    969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
    977 007400                                  LIST
    978
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 49-1
DEFINITIONS     ERRORS

    979 007400                          MOPINS  $ABS,-12
    992 007400                                  LIST
    993
    994
    995 007400                          MOPINS  FACT,22,T.SOP,.FLOAT+CS.BOOL
   1002 007400                                  LIST
   1003
   1004
   1005 007400                          MOPINS  NOT,50
   1031 007400                                  LIST
   1032
   1033
   1034 007400                          MOPINS  ROLL,100
   1057 007400                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

```
APL010  MACRC M1113  08-FEB-80 10:35  PAGE 51
DEFINITIONS     ERRORS

   1189 007400                          MOPINS  SHAP,102
   1211 007400                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 52
DEFINITIONS     ERRORS

   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 54
DEFINITIONS     ERRORS

   1701 007400                          DOPINS  1COM,124
   1709 007400                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 007400                          DOPINS  COMP,60,T.DIM
   1789 007400                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

```
   2151 007400                          MOPINS  1REV,146
   2160 007400                                  LIST
   2161
   2162
   2163
   2164
   2165
   2166 007400                          MOPINS  REVS,144,T.DIM
   2204 007400                                  LIST
```

```
   2206 007400                          DOPINS  1ROT,146
   2214 007400                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 007400                          DOPINS  ROT,144,T.DIM
   2328 007400                                  LIST
```

```
                                        MOPINS  TRAN,126
                                        .SBTTL  MIXED FUNCTION          MONADIC TRANSPOSE

                                        MOV     MONBLK,R2               ;Point to argument
                                        MOVB    A.DIMS(R2),R3           ;Get the number of dimensions
                                        CMP     R3,#1                   ;Is there more than one dimension?
                                        BGT     1$                      ;Yes?
                                        MOV     R2,R4                   ;No, Argument is a scalar or vector
                                        JMP     SAME                    ;So the result is the same as the argument

                                1$:     MOV     #A.DIM1,R4              ;Compute the size of the header
                                        ADD     R3,R4                   ;Plus Number of Dimensions
                                        ADD     R3,R4                   ;  (Word Offsets)
                                        MOV     A.ALEN(R2),R0           ;Get the size of the data
                                        MOV     MONTYP,R1               ;Get the Source type
                                        MOV     R1,RESTYP               ;Now it's the Result type
                                        ASH     SIZTAB(R1),R0           ;Convert data size to bytes
                                        MOV     LENTAB(R1),DEVIL        ;Get the Item Size
                                        MUL     #5,R1                   ;Type times 5 to index fetch routine tables
                                        MOV     LARGCT(R1),LARGCV       ;Setup Left argument fetch routine
                                        MOV     RESCT(R1),RESCVT        ;Setup Result store routine
                                        MOV     #SYSERR,RARGCV          ;Just in case
                                        ADD     R4,R0                   ;Add in the header size
                                        CALL    GETFRA                  ;Get result, and build the SI frame for it
                                        BITB    #A.NUM,A.BITS(R2)       ;Is the type numeric?
                                        BNE     2$                      ;Yes
                                        BIC     #T.NUM,SI.BIT(SI)       ;No, It is literial
                                2$:     MOV     R0,R1                   ;Copy the result pointer
                                        MOV     R3,OPTMP0               ;Save the number of dimensions
                                        ADD     R4,(R1)+                ;Add in offset
                                        TST     (R2)+                   ;Increment Pointer to Source
                                        MOV     (R2)+,(R1)              ;Bits & Rank
                                        BIC     #^C<A.ASB+A.NUM>*400,(R1)+  ;Clear Garbage
                                        MOV     (R2)+,R0                ;Save ALEN for later
                                        MOV     R0,(R1)+                ;ALEN
                                        ADD     R3,R2                   ;Point to end
                                        ADD     R3,R2                   ;  of source dimensions
                                        MOV     SP,OPTMP1               ;Save SATAN frame pointer
                                        MOV     SP,R4                   ;Copy it
                                        MOV     R3,OPTMP2               ;Copy the number of dimensions
                                        MOV     DEVIL,R3                ;Get Size of one element
                                ;
                                ;   Build SATAN stack frames
                                ;
                                3$:     PUSH    R1                      ;Save a register today
                                        MOV     R0,R1                   ;Copy the current result pointer offset
                                        CLR     R0                      ;Setup for divide
                                        DIV     -(R2),R0                ;Divide by current dimension size
                                        MOV     R0,R1                   ;Copy the new result pointer offset
                                        ASH     MONSFT,R1               ;Convert it to bytes
                                        MOV     R1,OPTMP3               ;Save it for later
                                        POP     R1                      ;Get back the register
                                                                        ;Build a SATAN frame
                                        CLR     -(SP)                   ;Result pointer
                                        MOV     OPTMP3,-(SP)            ;Result pointer offset
                                        CLR     -(SP)                   ;Source pointer
                                        MOV     (R2),-(SP)              ;Count limit
```

```
2387 00030  005046                      CLR     -(SP)           ;Count
2388 00030  010346                      MOV     R3,-(SP)        ;Source pointer offset
2389
2390 000304 011221                      MOV     (R2),(R1)+      ;Transposed dimension
2391 000306                             MUL     (R2),R3         ;Accumulate size for offset
2392 000320 005337  005072'             DEC     OPTMP2          ;Count this dimension
2393 000324 003336                      BGT     3$              ;Loop for all the dimensions
2394
2395 000326 013700  003120'             MOV     MONBLK,R0       ;Point to argument array block
2396 000332 005760  000004              TST     A.ALEN(R0)      ;Is it a Null array?
2397 000336 001420                      BEQ     5$              ;Yes, Then don't feed it to Satan
2398
2399 000340 013746  003152'             MOV     RESCVT,-(SP)    ;Final SATAN frame, Result routine pointer
2400 000344 005046                      CLR     -(SP)           ;Result pointer offset
2401 000346 005046                      CLR     -(SP)           ;Source pointer
2402 000350 012746  000001              MOV     #1,-(SP)        ;This last frame forces a copy operation
2403 000354 005046                      CLR     -(SP)           ;Count
2404 000356 005046                      CLR     -(SP)           ;Source pointer offset
2405
2406 000360 116000  000002              MOVB    A.DIMS(R0),R0   ;Get the number of dimensions
2407 000364 060002                      ADD     R0,R2           ;Point to the first source element
2408 000366 060002                      ADD     R0,R2           ;  all the way there
2409 000370 013704  005070'             MOV     OPTMP1,R4       ;Point to top level SATAN frame
2410
2411 000374                             CALL    SATAN           ;SATAN does the transpose
2412 000400 013706  005070'     5$:     MOV     OPTMP1,SP       ;Scrap the SI frames
2413 000404                             RETURN                  ;And Return
2415 000406                             LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 60
MIXED FUNCTION          MONADIC TRANSPOSE

   2417 000406                          DOPINS  TRAN,126
   2654 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 61
MIXED FUNCTION          MONADIC TRANSPOSE

   2656 000406                          DOPINS  ENCD,76
   2856 000406                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 62
MIXED FUNCTION          MONADIC TRANSPOSE

   2858 000406                          DOPINS  DECD,64
   2988 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 63
MIXED FUNCTION          MONADIC TRANSPOSE

 2990 000406                            MOPINS  GDWN,136,T.DIM
 3001 000406                                    LIST
 3002
 3003
 3004 000406                            MOPINS  GDUP,140,T.DIM
 3219 000406                                    LIST
```

```
   3221 000406                          MOPINS  MMD,122
   3316 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 65
MIXED FUNCTION          MONADIC TRANSPOSE

   3318 000406                          DOPINS  DMD,122
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 66
MIXED FUNCTION          MONADIC TRANSPOSE

   3864 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 67
MIXED FUNCTION          MONADIC TRANSPOSE

   3866 000406                          DOPINS  MEM,70
   3921 000406                                  LIST
```

```
 3923 000406                            MOPINS  NUB,106
 3986 000406                                    LIST
 3987
 3988
 3989
 3990
 3991
 3992 000406                            DOPINS  UNON,106
 4022 000406                                    LIST
 4023
 4024
 4025
 4026
 4027
 4028 000406                            DOPINS  INTR,66
 4042 000406                                    LIST
 4043
 4044
 4045
 4046
 4047
 4048 000406                            DOPINS  SDIF,50
 4161 000406                                    LIST
```

```
   4163 000406                          DOPINS  SCON,234
   4179 000406                                  LIST
   4180
   4181
   4182 000406                          DOPINS  CONT,112
   4200 000406                                  LIST
   4201
   4202
   4203 000406                          DOPINS  PSUB,236
   4209 000406                                  LIST
   4210
   4211
   4212 000406                          DOPINS  SUBS,116
   4288 000406                                  LIST
```

MIXED FUNCTION MONADIC TRANSPOSE

```
4290 000406                          MOPINS  EXE,70
4336 000406                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 71
MIXED FUNCTION          MONADIC TRANSPOSE

   4338 000406                          MOPINS  ENCD,76
   4411 000406                                  LIST
```

MIXED FUNCTION MONADIC TRANSPOSE

```
 4413 000406                    MOPINS  FMT,160
 4423 000406                            LIST
 4424
 4425
 4426
 4427
 4428
 4429 000406                    DOPINS  FMT,160
```

APLO10 MACRO M1113 08-FEB-80 10:35 PAGE 73
MIXED FUNCTION MONADIC TRANSPOSE

4953 000406 LIST

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 74
MIXED FUNCTION          MONADIC TRANSPOSE

 4955 000406                            MOPINS  IBM,132,,,MONIBM
 4958 000406                                    LIST
 4959
 4960
 4961 000406                            DOPINS  IBM,132
 5046 000406                                    LIST
```

MIXED FUNCTION MONADIC TRANSPOSE

```
5048 000406                        MOPINS  QUAD,74,T.NIL+T.SPOP
5075 000406                                LIST
5076
5077
5078
5079
5080
5081 000406                        DOPINS  QUAD,74,T.NIL+T.SPOP
5097 000406                                LIST
5098
5099
5100
5101
5102
5103 000406                        MOPINS  QQAD,142,T.NIL+T.SPOP
5135 000406                                LIST
5136
5137
5138
5139
5140
5141 000406                        DOPINS  QQAD,142,T.NIL+T.SPOP
5152 000406                                LIST
5153
5154
5155
5156
5157
5158 000406                        MOPINS  DQAD,134,T.NIL+T.SPOP
5192 000406                                LIST
5193
5194
5195
5196 000406                        DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 000406                                LIST
```

```
5201 000406                            MOPINS  BXAV,162,T.NIL
5219 000406                                    LIST
5220
5221
5222
5223
5224
5225 000406                            MOPINS  BXCT,164,T.NIL+T.SPOP
5234 000406                                    LIST
5235
5236
5237 000406                            DOPINS  BXCT,164,T.NIL+T.SPOP
5272 000406                                    LIST
5273
5274
5275
5276
5277
5278 000406                            MOPINS  BXIO,166,T.NIL+T.SPOP
5287 000406                                    LIST
5288
5289
5290 000406                            DOPINS  BXIO,166,T.NIL+T.SPOP
5302 000406                                    LIST
5303
5304
5305
5306
5307 000406                            MOPINS  BXLC,212,T.NIL
5340 000406                                    LIST
5341
5342
5343
5344
5345
5346
5347 000406                            MOPINS  BXPP,170,T.NIL+T.SPOP
5356 000406                                    LIST
5357
5358
5359 000406                            DOPINS  BXPP,170,T.NIL+T.SPOP
5381 000406                                    LIST
5382
5383
5384
5385
5386
5387
5388 000406                            MOPINS  BXPW,172,T.NIL+T.SPOP
5397 000406                                    LIST
5398
5399
5400 000406                            DOPINS  BXPW,172,T.NIL+T.SPOP
5417 000406                                    LIST
5418
5419
5420
```

```
   5421
   5422
   5423 000406                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 000406                                  LIST
   5433
   5434
   5435 000406                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 000406                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 000406                          MOPINS  BXTS,200,T.NIL
   5562 000406                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 000406                          MOPINS  BXWA,176,T.NIL
   5592 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 77
MIXED FUNCTION          MONADIC TRANSPOSE

   5594 000406                          MOPINS  BXCR,206
   5716 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 78
MIXED FUNCTION          MONADIC TRANSPOSE

   5718 000406                          MOPINS  BXEX,202
   5775 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 79
MIXED FUNCTION          MONADIC TRANSPOSE

   5777 000406                          MOPINS  BXFX,210
   5879 000406                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 80
MIXED FUNCTION          MONADIC TRANSPOSE

   5881 000406                          MOPINS  BXNC,204
   5950 000406                                  LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 81
MIXED FUNCTION          MONADIC TRANSPOSE

   5952 000406                          MOPINS  BXNL,214
   5963 000406                                  LIST
   5964
   5965
   5966 000406                          DOPINS  BXNL,214
   6104 000406                                  LIST
```

```
   6106 000406                          DOPINS  FGET,150,T.DIM
   6330 000406                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 000406                          DOPINS  FPUT,152,T.DIM
   6485 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 83
MIXED FUNCTION          MONADIC TRANSPOSE

  6487 000406                           DOPINS  FPNT,154,T.DIM
  6714 000406                                   LIST
```

```
  6716 000406                          MOPINS  1SCN,130
  6723 000406                                  LIST
  6724
  6725
  6726 000406                          MOPINS  SCAN,62,T.DIM
  6732 000406                                  LIST
  6733
  6734
  6735 000406                          MOPINS  1RDT,124
  6741 000406                                  LIST
  6742
  6743
  6744 000406                          MOPINS  RDCT,60,T.DIM
  7044 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 85
MIXED FUNCTION          MONADIC TRANSPOSE

   7046 000406                          MOPINS  GETIDN,0
   7193 000406                                  LIST
```

```
APL010  MACRO M1113  08-FEB-80 10:35  PAGE 86
MIXED FUNCTION          MONADIC TRANSPOSE

   7195 000406                          MOPINS  SATAN,0
   7404 000406                                  LIST
```

```
                                        MOPINS  OUTR,156
                                        .SBTTL  OPERATORS                       OUTER PRODUCT

                                                MOV     2*SI.LEN+SI.VAL(SI),R3  ;Get pointer to right argument
                                                MOV     SI.LEN+SI.VAL(SI),R4    ;Get the function number
                                                CALL    PRDSET                  ;Setup parameters for outer product
                                        ;+
                                        ;  Get space for the result
                                        ;-
                                                MOV     LARGBL,R2               ;LEFT ARG POINTER
                                                MOV     RARGBL,R3               ;RIGHT ARG POINTER
                                                MOVB    A.DIMS(R2),R1           ;GET # OF DIMS ON LEFT
                                                MOVB    A.DIMS(R3),R0           ;# OF DIMS ON RIGHT
                                                ADD     R0,R1                   ;# OF DIMS IN RESULT
                                                CMP     #DIMAX,R1               ;TOO MANY?
                                                BLE     2$                      ;YUP
                                                MOV     R1,R0                   ;COPY DIM COUNT
                                                CONINC  R0                      ;INC FOR SCALAR
                                                ASL     R0                      ;CONVERT FOR WORDS
                                                ADD     #A.DIM1,R0              ;GET HEADER SIZE
                                                PUSH    <R1,R0>                 ;# OF DIMS, OFFSET TO DATA
                                                MOV     A.ALEN(R2),R1           ;LENGTH OF LEFT ARG
                                                MOV     R1,LARGLN               ;SAVE
                                                MOV     A.ALEN(R3),R4           ;LENGTH OF RIGHT ARG
                                                MOV     R4,RARGLN               ;SAVE
                                                MUL     R4,R1                   ;LENGTH OF RESULT
                                                BCC     3$                      ;TOO LONG, ERROR
                                        1$:     WSFULL                          ;No Space Left
                                        2$:     DOMERR                          ;Too many dimensions
                                        3$:     PUSH    R1                      ;Save Length (ALEN)
                                                ASH     RESSFT,R1               ;Get Size in Bytes
                                                BVS     1$                      ;TOO BIG
                                                POP     R4                      ;Restore "ALEN"
                                                ADD     R1,R0                   ;TOTAL LENGTH OF RESULT
                                                CALL    GETFRA                  ;Get result, SI frame for it
                                                MOV     R0,RESULT               ;Save pointer to result block
                                                ADD     (SP)+,(R0)+             ;ADD IN OFFSET TO DATA
                                                MOVB    (SP)+,(R0)+             ;# OF DIMS IN RESULT
                                                MOVB    RESTYP,(R0)             ;GET RESULT TYPE
                                                BISB    #A.ASB,(R0)+            ;FLAG AS DATA ARRAY
                                                MOV     R4,(R0)+                ;ALEN
                                                MOV     #1,(R0)                 ;IF SCALAR, HACK IT
                                                MOVB    A.DIMS(R2),R4           ;# OF LEFT DIMS
                                                BEQ     7$                      ;NO...
                                                ADD     #A.DIM1,R2              ;POINT TO LEFT DIM LIST
                                        6$:     MOV     (R2)+,(R0)+             ;COPY DIM LIST
                                                SOB     R4,6$

                                        7$:     MOVB    A.DIMS(R3),R4           ;# OF RIGHT DIMS
                                                BEQ     9$                      ;NOTHING BUT A SCALAR
                                                ADD     #A.DIM1,R3              ;POINT TO DIM LIST
                                        8$:     MOV     (R3)+,(R0)+             ;COPY DIM LIST
                                                SOB     R4,8$

                                        9$:     MOV     RESULT,R1               ;POINT TO RESULT
                                                TST     A.ALEN(R1)              ;ANYTHING THERE?
                                                BEQ     13$                     ;NO, STOP THE WAR NOW!
```

```
7463 000662  011101                          MOV     (R1),R1          ;POINT TO HONEST-TO-GOD DATA
7464 000664  013702  003164'                 MOV     LARGPT,R2        ;POINT TO LEFT ARG
7465 000670  013704  005116'                 MOV     BLZBUB,R4        ;GET OP NUMBER
7466 000674                                  CALL    DOUTER           ;Now do the outer product
7467 000700  016502  177776          13$:    MOV     SI.VAL(SI),R2    ;SAVE POINTER TO RESULT
7468 000704  062705  177772                  ADD     #SI.LEN,SI       ;BACK UP A FRAME
7469 000710                                  CALL    KILL             ;ALL DONE.  KILL LEFT ARG
7470 000714  062705  177772                  ADD     #SI.LEN,SI       ;KILL OP
7471 000720  016504  177772                  MOV     SI.RET(SI),R4    ;SAVE RETURN
7472 000724                                  CALL    KILL             ;KILL RIGHT ARG
7473 000730                                  SIFRAM  R2,,R4           ;Restore Return, Stack bits, Pointer
7474 000740                                  RETURN                   ;NOW CONTINUE TO CONTINUE
7476 000742                                  LIST
```

```
 7478 000742                    MOPINS  DOUTER,0
 7504 000742                            LIST
 7505
 7506
 7507
 7508
 7509
 7510 000742                    MOPINS  PRDSET,0
 7564 000742                            LIST
 7565
 7566
 7567
 7568
 7569
 7570 000742                    MOPINS  INSPAC,0
 7612 000742                            LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 89
OPERATORS                OUTER PRODUCT

   7614 000742                          DOPINS  INNR,156
```

```
7960 000742                                    LIST
```

```
   7962 000742                          MOPINS  SUBR,0
```

```
   8725 000742                                    LIST
```

```
8727 000742                            MOPINS  RAND,0
8728                                   .SBTTL  SUBROUTINES              .RAND   RANDOM NUMBER GENERATOR
8729
8730                                   ;++
8731                                   ; FUNCTIONAL DESCRIPTION:
8732                                   ;       GENERATES A RANDOM INTEGER IN THE RANGE OF .IO AC0.  RETURNS WITH
8733                                   ;       INTEGER IN AC0.  RANDOM SEQUENCE COEFFICIENTS SELECTED ACCORDING
8734                                   ;       TO KNUTH, 'SEMI-NUMERICAL ALGORITHMS''.
8735                                   ;       NOTE:  THIS ROUTINE HAS BEEN TUNED TO GIVE A VERY LONG CYCLE OF NUMBERS.
8736                                   ;--
8737
8738 000742                                    .PSECT  CODE
8739
8740 000742                            .RAND::
8741 000742                                    STF     AC1,-(SP)       ;SAVE AC1 FOR CALLER
8742 000750                                    PUSH    R1              ;AND R1
8743 000752  013701  003062'                   MOV     RANDY,R1        ;GET RANDOM SEED
8744 000756                                    MUL     #14525.,R1      ;TIMES MAGIC FACTOR (14525 MOD 8 = 5)
8745 000772  062701  033031                    ADD     #13849.,R1      ;PLUS CONSTANT (1/2-SQR(3)/6)*2^16
8746 000776  010137  003062'                   MOV     R1,RANDY        ;SAVE SEED
8747 001002  000301                            SWAB    R1              ;CLEAR CARRY, SWAP SIGNIFICANT DIGITS
8748 001004  006001                            ROR     R1              ;MAKE POSITIVE
8749 001006  010137  003064'                   MOV     R1,RANDAC       ;AND SAVE AS FIRST WORD OF NUMBER
8750 001012                                    MUL     RANDAC+6,R1     ;GET LAST WORD OF RANDOM NUMBER
8751 001026                                    MUL     #29813.,R1      ;TIMES ANOTHER MAGIC FACTOR
8752 001042  062701  033031                    ADD     #13849.,R1      ;PLUS TRUSTY CONSTANT
8753 001046  010137  003066'                   MOV     R1,RANDAC+2;    ;SAVE NEW SECOND WORD
8754 001052                                    MUL     #29813.,R1      ;CYCLE AGAIN
8755 001066  062701  033031                    ADD     #13849.,R1
8756 001072  010137  003070'                   MOV     R1,RANDAC+4     ;TO GET THIRD WORD
8757 001076                                    MUL     #29813.,R1      ;ONCE MORE
8758 001112  062701  033031                    ADD     #13849.,R1      ;FOR LAST WORD
8759 001116  010137  003072'                   MOV     R1,RANDAC+6
8760 001122                                    LDF     RANDAC,AC1      ;LOAD THE RANDOM NUMBER
8761 001132                                    LDEXP   #1,AC1          ;1 <= AC1 < 2
8762 001142                                    SUBF    #^F1.0,AC1      ;0 <= AC1 < 1
8763 001152                                    MULF    AC1,AC0         ;0 <= AC0 < ARG
8764 001160                                    MODF    #^F1.0,AC0      ;MAKE INTEGER
8765 001170                                    STF     AC1,AC0         ;RETURN INTEGER PART
8766 001176                                    ADDF    FORIGN,AC0      ;SCALE BY ORIGIN
8767 001206                                    POP     R1              ;RESTORE R1
8768 001210                                    LDF     (SP)+,AC1       ;RESTORE AC1
8769 001216                                    RETURN
8771 001220                                    LIST
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 94
SUBROUTINES                .RAND   RANDOM NUMBER GENERATOR

      1          000001                              .END
```

APLO10 MACRO M1113 08-FEB-80 10:35 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DECD= 000002    EIS    = 000000
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DIV = 000000    ELESIZ 005132R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DMD = 000010    ENDCOR 000004R     025
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DQAD= 177777    EOF    = 104475
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.ENCD= 000002    ERRNUM 004772R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.EQ  = 000000    ERRPTR 004770R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.EXPD= 000002    ERRO   004774R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FGET= 000005    EXEBUF 005054R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FMT = 000004    E$CT   005056R     025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPNT= 000005    E$PT   005060R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.FPUT= 000005    FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GE  = 000000    FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.GT  = 000000    FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IBM = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.IND = 000002    FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INNR= 000013    FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.INTR= 000006    FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LE  = 000000    FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LOG = 000001    FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.LT  = 000000    FA.RD = 000001
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MAX = 000000    FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MEM = 000002    FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MIN = 000000    FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.MUL = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.NE  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NN  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.NR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.OR  = 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PLUS= 000000    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    DEBUG = 000000    D.PSUB= 000006    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.PWR = 000001    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R     025 D.QQAD= 000000    FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.QUAD= 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R     025 D.RES = 000000    FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIMAX = 000020    D.RHO = 000000    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R     025 D.ROT = 000003    FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SCON= 000006    FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010    DIMTYP  003116R     025 D.SDIF= 000006    FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466    D.SUB = 000000    FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R     025 DOUTER= ****** GX D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R     025 DYDCNT  003224R     025 D.UNON= 000006    FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R     025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R     025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200          D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010          D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020          D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040          D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110           CS.ONE= 000100          D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R     025 D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105           CTMAX   000212R     025 D.COMP= 000001    EDICHR  005063R     025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R     025 D.CONT= 000006    EDICNT  005064R     025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162          D.CTLM= 000000    EDIFLG  005062R     025 FHEADR  005012R     025
CHRFRO= 000135           C.BLOK= 000064          D.DEAL= 000003    EDIT  = 000003          FHRET   005040R     025
```

SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          IASH  = ****** GX       MACHIN= 000106          M.QUAD= 000000
FIS   = 000000          F.FNDF= 000010          ICHAN   005050R     025 MAXCHN= 000014          M.RAND= 000012
FISCON= 046000          F.FTYP= 000116          IDIV  = ****** GX       MONBLK  003120R     025 M.RAVL= 000000
FIXPTR  003074R     025 F.FUN = 000010          ILFNAM= 104464          MONCVT  003132R     025 M.RCIP= 000000
FLAGS   004764R     025 F.FVER= 000120          ILLCHN= 104540          MONLEN  003130R     025 M.RDCT= 000011
FLCBUF  003250R     025 F.HEAD= 000006          ILLSW = 104565          MONPTR  003122R     025 M.REVS= 000003
FLINE   005014R     025 F.HIBK= 000004          IMUL  = ****** GX       MONSFT  003126R     025 M.ROLL= 000000
FLOEXP  003270R     025 F.LBL = 000020          INDERR= 104410          MONTYP  003124R     025 M.SATA= 000001
FLONEG  003272R     025 F.LINE= 000012          INNTMP  003250R     025 MONVAL  003134R     025 M.SCAN= 000011
FLSWTB  000222R     025 F.LOC = 000004          INNTM2  005016R     025 M$OUTR  000406RG    020 M.SHAP= 000000
FLTLEN= 000010          F.LSTC= 000004          INPTR   000064R     025 M$TRAN  000000RG    020 M.SIGN= 000000
FNINSI  005044R     025 F.LUN = 000042          INSNAM= 104462          M.ABS = 000000          M.SUBR= 000001
FORIGN  000032R     025 F.MBCT= 000054          IOERR = 104477          M.BOL = 000200          M.SYM = 000203
FO.APD= 000106          F.MBC1= 000055          I.CNT = 177776          M.BXAV= 000007          M.SYN = 000201
FO.MFY= 000002          F.MBFG= 000056          I.DIM = 177772          M.BXCR= 000007          M.TRAN= 000012
FO.RD = 000001          F.MON = 000002          I.LEN = 000012          M.BXCT= 000007          M.$ABS= 000000
FO.UPD= 000006          F.NIL = 000001          I.OERR= 000062          M.BXEX= 000007          M.$CEI= 000000
FO.WRT= 000016          F.NRBD= 000024          I.OLD = 177770          M.BXFX= 000007          M.$FLR= 000000
FPMODE  000042R     025 F.NREC= 000030          I.PTR = 177774          M.BXIO= 000007          M.$IDN= 000000
FPP   = 000000          F.OPN = 000010          I.STR = 177766          M.BXLC= 000003          M.$NEG= 000000
FPPLEN= 000004          F.OVBS= 000030          KILL  = ****** GX       M.BXNC= 000007          M.$SIG= 000000
FREBLK  007400R     025 F.PREV= 000000          LABERR= 104406          M.BXNL= 000007          M.1RDT= 000011
FUZZ    000142R     025 F.PRNT= 100000          LABLPT  005042R     025 M.BXPP= 000007          M.1REV= 000003
FXPTR1  003076R     025 F.PROT= 010000          LARGBL  003162R     025 M.BXPW= 000007          M.1SCN= 000011
F$EVAL= 000100          F.RACC= 000016          LARGCT= ****** GX       M.BXRL= 000007          NAMBUF  003240R     025
F$EVI = 000200          F.RATT= 000001          LARGCV  003176R     025 M.BXTS= 000007          NB.DEV= 000200
F$EXEC= 000400          F.RCNM= 000034          LARGFL  003160R     025 M.BXWA= 000007          NB.DIR= 000100
F$FUNC= 001000          F.RCTL= 000017          LARGLE  003174R     025 M.CEI = 177777          NB.NAM= 000004
F.ACTL= 000076          F.READ= 004000          LARGLN  003166R     025 M.CHR = 000202          NB.SD1= 000400
F.ALOC= 000040          F.REDF= 000040          LARGPT  003164R     025 M.CMT = 000206          NB.SD2= 001000
F.BBFS= 000062          F.RES = 000040          LARGSF  003172R     025 M.DOUT= 000006          NB.SNM= 000040
F.BDB = 000070          F.RET = 177772          LARGTY  003170R     025 M.DQAD= 000006          NB.STP= 000020
F.BGBC= 000057          F.RSIZ= 000002          LARGVA  003200R     025 M.ENCD= 000004          NB.SVR= 000010
F.BITS= 177774          F.RTYP= 000000          LBINC   005100R     025 M.EXE = 000000          NB.TYP= 000002
F.BKDN= 000026          F.R2  = 000014          LBUFF   003316R     025 M.EXP = 000001          NB.VER= 000001
F.BKDS= 000020          F.R3  = 000016          LBUFLN= 000621          M.FACT= 000001          ND.DEV= 000200
F.BKEF= 000050          F.R4  = 000020          LCOUNT  005106R     025 M.FIL = 000177          ND.EXT= 000002
F.BKP1= 000051          F.SEQN= 000100          LENERR= 104412          M.FLR = 177777          ND.NAM= 000004
F.BKST= 000024          F.SPDV= 000072          LENTAB  000054R     025 M.FMT = 000004          ND.PPN= 000100
F.BKVB= 000064          F.SPUN= 000074          LEXPTR  003100R     025 M.GDUP= 000003          ND.$EX= 000000
F.BR  = 000020          F.STBK= 000036          LFTOP   005066R     025 M.GDWN= 000003          ND.$ES= 000020
F.CHR = 000075          F.STK = 000022          LINC    005102R     025 M.GETI= 000006          ND.$NM= 000000
F.CNTG= 000034          F.STOP= 000004          LINE    005016R     025 M.IBM = 000003          ND.$NS= 000040
F.CODE= 000010          F.STP = 000001          LININC  005026R     025 M.IDN = 000000          ND.$PN= 001400
F.DFNB= 000046          F.SUSP= 000002          LISIZE= 000020          M.INSP= 000006          NEGSGN  005063R     025
F.DSPT= 000044          F.TRC = 000002          LITOP   005072R     025 M.IOTA= 000000          NINPT1  003302R     025
F.DVNM= 000134          F.UNIT= 000136          LNLNG = 104541          M.LIT = 000205          NINPT2  003300R     025
F.EFBK= 000010          F.URBD= 000020          LOCLCT  005036R     025 M.LN  = 000001          NINSTK  003304R     025
F.EFN = 000050          F.VAL = 177776          LOWCOR  000002R     025 M.MMD = 000010          NINTMP  003310R     025
F.END = 002000          F.VBN = 000064          LSTERP  004766R     025 M.NEG = 000000          NINTYP  003306R     025
F.EOBB= 000032          F.VBSZ= 000060          LWIDTH  000070R     025 M.NOT = 000000          NOTAVL= 104472
F.ERR = 000052          GETFRA= ****** GX       L.BITS= 000003          M.NUB = 000006          NOTOPN= 104473
F.FACC= 000043          GNOME   003102R     025 L.CNT = 000002          M.NUM = 000204          NPRT1   004762R     025
F.FFBY= 000014          HASHC   005136R     025 L.PRNT= 000001          M.OUTR= 000012          NULL    007364R     025
F.FLAG= 000002          HASHTB  005140R     025 L.PTR = 000000          M.PITM= 000000          NUMCHN= 000020
F.FLB = 000100          HSHMSK= 177601          L.PTR1= 000004          M.PRDS= 000006          NUMDFL  003230R     025
F.FNAM= 000110          HSHSIZ= 000200          L.SUBS= 000002          M.QQAD= 000000          NUMEFL  003233R     025
```

APLO10 MACRO M1113 08-FEB-80 10:35 PAGE 94-3
SYMBOL TABLE

```
NUMESN  003236R  025 PRECSN  000052R  025 R.NEXT= 000006       SY.CLR= 000040       T.OPFN= 000001
NUMFLG  003230R  025 RANDAC  003064R  025 R.OFF = 000000       SY.ERP= 000020       T.OTHR= 040000
NUMLEN  000062R  025 RANDY   003062R  025 R.PREV= 000004       SY.NOU= 000100       T.SOP = 000002
NUMNEG  003232R  025 RANERR= 104411           R.PTR = 000000   SY.OPN= 000002       T.SPOP= 000040
NUMQFL  003234R  025 RARGBL  003204R  025 R.SEQ = 000003       SY.PMT= 001000       T.TMP = 000200
NUMRIG  003231R  025 RARGCV  003220R  025 R.VAR = 000002       SY.SRT= 000400       T.VAL = 100000
NUMWID  003226R  025 RARGFL  003202R  025 R6    =%000006       SY.SVC= 000004       T.VAR = 001000
NUMZFL  003235R  025 RARGLE  003216R  025 R7    =%000007       SY.SVL= 000200       UPPER = 177400
NYIERR= 104432       RARGLN  003210R  025 SAME  = ****** GX    S.AB  = 177776       UTNAME  000344R  023
N.BLSZ= 000036       RARGPT  003206R  025 SATAN = ****** GX    S.ASB = 100000       VALERR= 104413
N.DESC= 000020       RARGSF  003214R  025 SI    =%000005       S.BFHD= 000020       VARBLE  007400R  025
N.DEV = 000032       RARGTY  003212R  025 SIDAMN= 104433       S.BITS= 000002       VARBLS  000000R  025
N.DEVN= 000034       RARGVA  003222R  025 SIFRAM  003314R  025 S.CHR1= 000003       VERNUM  000000R  025
N.DID = 000024       RBASE   005074R  025 SISTAK  006350R  025 S.FATT= 000016       VRSION= 000002
N.DVNM= 000032       RCOUNR  005112R  025 SITOP   006346R  025 S.FDB = 000140       WIDTH   000072R  025
N.EXT = 000014       RCOUNT  005110R  025 SITP    007144R  025 S.FLB = 040000       WROUND  000020R  025
N.FID = 000000       REDCNT  005070R  025 SIWARN= 000200       S.FNAM= 000006       WSDATE  000136R  025
N.FNAM= 000006       REDIGN  005066R  025 SIZTAB  000044R  025 S.FNB = 000036       WSFULL= 104403
N.FTYP= 000014       REDOFF  005072R  025 SI.BIT= 177774       S.FNBW= 000017       WSNAME  000074R  025
N.FVER= 000016       REENTR  003060R  025 SI.LEN= 177772       S.FNTY= 000004       WSTIME  000132R  025
N.NAME= 000006       REGTYP= 000001           SI.RET= 177772   S.FTYP= 000002       $ADDF = ****** GX
N.NEXT= 000022       RESBIT  003142R  025 SI.VAL= 177776       S.LEN = 000002       $FIS  = ****** G
N.PPN = 000024       RESCT = ****** GX        SKPSIZ  005134R  025 S.LINK= 177774    $FPP  = ****** G
N.STAT= 000020       RESCVT  003152R  025 SPHOLD  000004R  024 S.NFEN= 000020       $FPPLN= ****** G
N.UNIT= 000034       RESERR= 104440           SPINIT  000002R  024 S.TYPE= 140000    $LDEXP= ****** GX
OBUFLN= 001000       RESLEN  003150R  025 STKINC= 000774       S.VAL = 000000       $LDF  = ****** GX
OCHAN   005052R  025 RESPTR  003140R  025 STOP  = 104420       TKDIMS  005072R  025 $MACHN= ****** G
OMODE   000402R  023 RESSFT  003146R  025 SUBCNT  003104R  025 TKFRAM  005066R  025 $MODF = ****** GX
OPTMP0  005066R  025 RESTYP  003144R  025 SUBDCT  003106R  025 TKLEFT  005070R  025 $MULF = ****** GX
OPTMP1  005070R  025 RESULT  003136R  025 SUBPTR  003110R  025 TMODE   000403R  023 $RSTS = ****** G
OPTMP2  005072R  025 RESVAL  003154R  025 SYMB1   005344R  025 TMPTAG= 000166       $RSX  = ****** G
OPTMP3  005074R  025 RINC    005104R  025 SYMF    005340R  025 TTYBUF  004140R  025 $RT11 = ****** G
OPTMP4  005076R  025 RLTBAS  005066R  025 SYMPT1  005346R  025 TTYLEN= 000622       $STF  = ****** GX
OPTMP5  005100R  025 ROTINC  005070R  025 SYMTAB  005352R  025 TURKEY= 104436       $SUBF = ****** GX
OPTMP6  005102R  025 ROUND   000010R  025 SYMTBF= 104404       TYPE  = 000166       .BOOLE= 000002
OPTMP7  005104R  025 ROVER   003312R  025 SYMTOP  005350R  025 T.BOL = 020000       .CHARA= 000000
OPTMP8  005106R  025 RSTS  = 000000           SYMTP   006272R  025 T.CON = 004000    .DEF. = 000002
OPTMP9  005110R  025 RSX   = 000115           SYMWRN= 000054       T.DIM = 000100   .FLOAT= 000006
OPTM10  005112R  025 RTOP    005070R  025 SYNERR= 104407       T.FUN = 010000       .FSRCB= ****** G
OPTM14  005026R  025 RT11  = 000000           SYSERR= 104402   T.LIST= 000400       .INTEG= 000004
ORIGIN  000030R  025 R.BITS= 000003           SYSTAT  000000R  024 T.LIT = 000020    .MOLUN  000404R  023
OUTPTR  000066R  025 R.FIX = 000001           SY.BAD= 000001   T.MON = 000010       .RAND   000742RG 020
OVROPR= 000012       R.LCT = 000002           SY.BRK= 100000   T.NIL = 000004       ...GBL= 000000
PATCH   007344R  025 R.LN  = 000002           SY.BS = 000010   T.NUM = 002000       ...TPC= 000020
PRDSET= ****** GX    R.LNUM= 000010

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
```

```
APLO10  MACRO M1113  08-FEB-80 10:35  PAGE 94-4
SYMBOL TABLE

ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    001220      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13259 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:09
APLO10,DM1:[305,301]APLO10/-SP=APLRX1,APLDEF,APLO10,APLOP,END
```

FILEID**APL011

```
  AAAAAA      PPPPPPPP      LL              000000        11          11
  AAAAAA      PPPPPPPP      LL              000000        11          11
AA      AA    PP      PP    LL            00      00    1111        1111
AA      AA    PP      PP    LL            00      00    1111        1111
AA      AA    PP      PP    LL            00      00      11          11
AA      AA    PP      PP    LL            00      00      11          11
AA      AA    PPPPPPPP      LL            00      00      11          11
AA      AA    PPPPPPPP      LL            00      00      11          11
AAAAAAAAAA    PP            LL            00      00      11          11
AAAAAAAAAA    PP            LL            00      00      11          11
AA      AA    PP            LL            00      00      11          11      ....
AA      AA    PP            LL            00      00      11          11      ....
AA      AA    PP            LLLLLLLLLL      000000      111111      111111    ....
AA      AA    PP            LLLLLLLLLL      000000      111111      111111    ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 10:38
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    89- 7615    OPERATOR                 INNER PRODUCT
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 1


      1        000000                  RT11    =  0          ;NOT RT-11
      2        000000                  RSTS    =  0          ;NOT RSTS/E
      3        000115                  RSX     = 'M          ;RSX-11/M
      4
      5        000000                  EIS     =  0          ;NO EIS
      6        000000                  FIS     =  0          ;NO FIS
      7        000000                  FPP     =  0          ;NO FPP
      8
      9        000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 44
DEFINITIONS     ERRORS

      1                                 .TITLE  APLO11
      2                                 .IDENT  \V02.03\
      3
      4                                 ;THIS IS USED WHEN ASSEMBLING APLOP TO DEFINE A NEW OVERLAY SEGMENT
      5                                 ;CONTAINING OPERATOR ROUTINES.  IN THE COMMON DEFINITION FILE,
      6                                 ;'OVROPR' IS DEFINED AS ZERO.  IF IT IS DEFINED AS ONE BEFORE APLOP
      7                                 ;IS ASSEMBLED, ONLY THE CODE DESIGNATED FOR THE OPERATOR OVERLAY IS
      8                                 ;ASSEMBLED.  THEREFORE, THE COMMAND TO ASSEMBLE APLO11 IS:
      9                                 ;
     10                                 ;       APLO11 < ...,APLO11,APLOP
     11                                 ;
     12
     13          000013                OVROPR= 11.                     ;OVERLAY SPECIFIED OPERATORS
```

```
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       THIS MODULE CONTAINS THE CODE FOR MOST APL OPERATORS WHICH
                        ;       HAVE BEEN IMPLEMENTED SO FAR.  TWO MACROS ARE USED TO DEFINE
                        ;       OPERATOR ROUTINES.  'DOPINS' DEFINES A DYADIC OPERATOR, AND
                        ;       'MOPINS' DEFINES A MONADIC OPERATOR.  THE ARGUMENTS TO THE
                        ;       MACROS ARE THE NAME OF THE OPERATOR ROUTINE AND THE OPERATOR
                        ;       NUMBER FOR THE OPERATOR (WHICH CAN BE FOUND BY LOOKING IN
                        ;       'VTAB').  FOR DYADIC SCALAR OPS, THE LEFT ARG IS IN AC1, THE
                        ;       RIGHT ARG IS IN AC2, AND THE RESULT IS RETURNED IN AC0.  FOR
                        ;       MONADIC SCALAR OPS, THE ARGUMENT IS IN AC1, AND THE RESULT
                        ;       IS RETURNED IN AC0.  SCALAR OPS MUST LEAVE THEIR ARGUMENTS
                        ;       INTACT!
                        ;--



                        ;THESE MACROS ARE FOR OVERLAY CODE.   THEY ARE USED THE SAME AS
                        ;RESIDENT MACROS.  THE DIFFERENCE COMES IN THE WAY THEY WORK.  THE
                        ;'PRMRES' MACRO, FORCES ALL CODE BETWEEN IT AND THE NEXT '.IFTF'/'LIST'
                        ;TO NOT BE ASSEMBLED.   THE 'MOPINS' AND 'DOPINS' DON'T PUT ANYTHING IN
                        ;ANY TABLES.  OTHER THAN THAT THEY WORK BASICLY THE SAME AS THE RESIDENT
                        ;MACROS.

                        .MACRO  DOPINS NAME,NUM,BITS,BOZO,NOTME
                        .IIF    NDF D.'NAME, .ERROR ;D.'NAME IS NOT DEFINED BY OVROP!
                        .IF EQ OVROPR-D.'NAME
                                .PSECT  CODE
                        D$'NAME::
                        .ENDM

                        .MACRO  MOPINS  NAME,NUM,BITS,BOZO,NOTME
                        .IIF    NDF M.'NAME, .ERROR ;M.'NAME IS NOT DEFINED BY OVROP!
                        .IF EQ OVROPR-M.'NAME
                                .PSECT  CODE
                                .IF NE NUM
                        M$'NAME::
                            .ENDC
                        .ENDM

                        .MACRO  LIST
                                .ENDC
                        .ENDM

                        .MACRO  PRMRES
                                .IF EQ 1
                        .ENDM

                                        ;************ ASSEMBLING THE FILE!!!!!
007400                  PRMRES
                                LIST
```

```
    201 007400                          DOPINS  PLUS,2,T.SOP,,INTEGER+CS.IDN+CS.ASS
    208 007400                                  LIST
    209
    210 007400                          DOPINS  $PLU,-2
    224 007400                                  LIST
    225
    226 007400                          DOPINS  SUB,4,T.SOP,,INTEGER+CS.IDN
    233 007400                                  LIST
    234
    235 007400                          DOPINS  $SUB,-4
    249 007400                                  LIST
    250
    251
    252 007400                          DOPINS  MUL,6,T.SOP,,INTEGER+CS.BOOL+CS.IDN+CS.ONE+CS.ASS
    264 007400                                  LIST
    265
    266
    267 007400                          DOPINS  $MUL,-6
    284 007400                                  LIST
    285
    286 007400                          DOPINS  DIV,10,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    323 007400                                  LIST
    324
    325
    326 007400                          DOPINS  PWR,20,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    392 007400                                  LIST
    393
    394
    395 007400                          DOPINS  LOG,54,T.SOP,,FLOAT+CS.BOOL
    417 007400                                  LIST
    418
    419
    420 007400                          DOPINS  CIRC,52,T.SOP,,FLOAT
    477 007400                                  LIST
    478
    479
    480 007400                          DOPINS  MIN,14,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    490 007400                                  LIST
    491
    492 007400                          DOPINS  $MIN,-14
    516 007400                                  LIST
    517
    518
    519 007400                          DOPINS  MAX,16,T.SOP,,INTEGER+CS.BOOL+CS.ASS
    529 007400                                  LIST
    530
    531 007400                          DOPINS  $MAX,-16
    555 007400                                  LIST
    556
    557
    558 007400                          DOPINS  RES,12,T.SOP,,INTEGER+CS.BOOL+CS.IDN
    589 007400                                  LIST
    590
    591
    592 007400                          DOPINS  $RES,-12
    616 007400                                  LIST
    617
```

```
    618
    619 007400                          DOPINS  COMB,22,T.SOP,,FLOAT+CS.BOOL+CS.IDN+CS.ONE
    639 007400                                  LIST
```

```
641 007400                          DOPINS  AND,42,T.SOP,.BOOLEAN+CS.IDN+CS.ONE+CS.ASS
652 007400                                  LIST
653
654
655 007400                          DOPINS  OR,40,T.SOP,.BOOLEAN+CS.IDN+CS.ASS
662 007400                                  LIST
663
664
665 007400                          DOPINS  NN,46,T.SOP,.BOOLEAN
676 007400                                  LIST
677
678 007400                          DOPINS  NR,44,T.SOP,.BOOLEAN
687 007400                                  LIST
688
689
690 007400                          DOPINS  LT,24,T.LIT+T.SOP,.CHARACTER+CS.IDN
700 007400                                  LIST
701
702 007400                          DOPINS  LE,26,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
712 007400                                  LIST
713
714
715 007400                          DOPINS  EQ,30,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
733 007400                                  LIST
734
735 007400                          DOPINS  GE,36,T.LIT+T.SOP,.CHARACTER+CS.IDN+CS.ONE
745 007400                                  LIST
746
747
748 007400                          DOPINS  GT,32,T.LIT+T.SOP,.CHARACTER+CS.IDN
758 007400                                  LIST
759
760
761 007400                          DOPINS  NE,34,T.LIT+T.SOP,.CHARACTER+CS.IDN
780 007400                                  LIST
781
```

```
    783 007400                          MOPINS  FLR,14,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    786 007400                                  LIST
    787
    788 007400                          MOPINS  CEI,16,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT,M$IDN
    791 007400                                  LIST
    792
    793 007400                          MOPINS  IDN,2,T.SOP,,INTEGER+CS.BOOL
    799 007400                                  LIST
    800
    801
    802 007400                          MOPINS  $IDN,-2
    808 007400                                  LIST
    809
    810
    811 007400                          MOPINS  $FLR,-14
    855 007400                                  LIST
    856
    857
    858 007400                          MOPINS  $CEI,-16
    868 007400                                  LIST
    869
    870
    871 007400                          MOPINS  NEG,4,T.SOP,,INTEGER
    878 007400                                  LIST
    879
    880 007400                          MOPINS  $NEG,-4
    887 007400                                  LIST
    888
    889
    890 007400                          MOPINS  SIGN,6,T.SOP,,INTEGER+CS.BOOL+CS.FLOAT
    901 007400                                  LIST
    902
    903 007400                          MOPINS  $SIG,-6
    914 007400                                  LIST
    915
    916
    917 007400                          MOPINS  RCIP,10,T.SOP,,FLOAT+CS.BOOL
    928 007400                                  LIST
    929
    930 007400                          MOPINS  EXP,20,T.SOP,,FLOAT
    939 007400                                  LIST
    940
    941
    942 007400                          MOPINS  LN,54,T.SOP,,FLOAT+CS.BOOL
    949 007400                                  LIST
    950
    951
    952 007400                          MOPINS  PITM,52,T.SOP,,FLOAT
    959 007400                                  LIST
    960
    961 007400                          PRMRES
    966 007400                                  LIST
    967
    968
    969 007400                          MOPINS  ABS,12,T.SOP,,INTEGER+CS.BOOL
    977 007400                                  LIST
    978
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 49-1
DEFINITIONS     ERRORS

     979 007400                         MOPINS  $ABS,-12
     992 007400                                 LIST
     993
     994
     995 007400                         MOPINS  FACT,22,T.SOP,.FLOAT+CS.BOOL
    1002 007400                                 LIST
    1003
    1004
    1005 007400                         MOPINS  NOT,50
    1031 007400                                 LIST
    1032
    1033
    1034 007400                         MOPINS  ROLL,100
    1057 007400                                 LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 50
DEFINITIONS     ERRORS

   1059 007400                          DOPINS  DEAL,100
   1187 007400                                  LIST
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 51
DEFINITIONS     ERRORS

   1189 007400                          MOPINS  SHAP,102
   1211 007400                                  LIST
   1212
   1213
   1214
   1215
   1216
   1217 007400                          DOPINS  RHO,102
   1313 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   1315 007400                          MOPINS  RAVL,56
   1325 007400                                  LIST
   1326
   1327
   1328
   1329
   1330
   1331 007400                          DOPINS  CTLM,56,T.DIM
   1596 007400                                  LIST
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 53
DEFINITIONS     ERRORS

   1598 007400                          MOPINS  IOTA,72
   1632 007400                                  LIST
   1633
   1634
   1635
   1636
   1637
   1638 007400                          DOPINS  IND,72
   1699 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 54
DEFINITIONS     ERRORS

   1701 007400                          DOPINS  1COM,124
   1709 007400                                  LIST
   1710
   1711
   1712
   1713
   1714
   1715 007400                          DOPINS  COMP,60,T.DIM
   1789 007400                                  LIST
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 55
DEFINITIONS     ERRORS

   1791 007400                          DOPINS  1EXP,130
   1799 007400                                  LIST
   1800
   1801
   1802
   1803
   1804
   1805 007400                          DOPINS  EXPD,62,T.DIM
   1864 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 56
DEFINITIONS     ERRORS

   1866 007400                          DOPINS  DROP,104
   1907 007400                                  LIST
   1908
   1909
   1910
   1911 007400                          DOPINS  TAKE,114
   2149 007400                                  LIST
```

DEFINITIONS ERRORS

```
2151 007400                          MOPINS  1REV,146
2160 007400                                  LIST
2161
2162
2163
2164
2165
2166 007400                          MOPINS  REVS,144,T.DIM
2204 007400                                  LIST
```

```
   2206 007400                          DOPINS  1ROT,146
   2214 007400                                  LIST
   2215
   2216
   2217
   2218
   2219
   2220 007400                          DOPINS  ROT,144,T.DIM
   2328 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 59
DEFINITIONS     ERRORS

   2330 007400                          MOPINS  TRAN,126
   2415 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   2417 007400                          DOPINS  TRAN,126
   2654 007400                                  LIST
```

C 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 61
DEFINITIONS     ERRORS

   2656 007400                          DOPINS  ENCD,76
   2856 007400                                  LIST
```

D 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 62
DEFINITIONS     ERRORS

   2858 007400                          DOPINS  DECD,64
   2988 007400                                  LIST
```

E 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 63
DEFINITIONS     ERRORS

   2990 007400                          MOPINS  GDWN,136,T.DIM
   3001 007400                                  LIST
   3002
   3003
   3004 007400                          MOPINS  GDUP,140,T.DIM
   3219 007400                                  LIST
```

F 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 64
DEFINITIONS     ERRORS

   3221 007400                          MOPINS  MMD,122
   3316 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 65
DEFINITIONS     ERRORS

   3318 007400                          DOPINS  DMD,122
```

H 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 66
DEFINITIONS     ERRORS

   3864 007400                                  LIST
```

I 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 67
DEFINITIONS     ERRORS

   3866 007400                          DOPINS  MEM,70
   3921 007400                                  LIST
```

J 10

DEFINITIONS ERRORS

```
3923 007400                          MOPINS  NUB,106
3986 007400                                  LIST
3987
3988
3989
3990
3991
3992 007400                          DOPINS  UNON,106
4022 007400                                  LIST
4023
4024
4025
4026
4027
4028 007400                          DOPINS  INTR,66
4042 007400                                  LIST
4043
4044
4045
4046
4047
4048 007400                          DOPINS  SDIF,50
4161 007400                                  LIST
```

K 10

DEFINITIONS   ERRORS

```
   4163 007400                          DOPINS  SCON,234
   4179 007400                                  LIST
   4180
   4181
   4182 007400                          DOPINS  CONT,112
   4200 007400                                  LIST
   4201
   4202
   4203 007400                          DOPINS  PSUB,236
   4209 007400                                  LIST
   4210
   4211
   4212 007400                          DOPINS  SUBS,116
   4288 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 70
DEFINITIONS     ERRORS

   4290 007400                          MOPINS  EXE,70
   4336 007400                                  LIST
```

M 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 71
DEFINITIONS     ERRORS

   4338 007400                          MOPINS  ENCD,76
   4411 007400                                  LIST
```

N 10

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 72
DEFINITIONS     ERRORS

   4413 007400                          MOPINS  FMT,160
   4423 007400                                  LIST
   4424
   4425
   4426
   4427
   4428
   4429 007400                          DOPINS  FMT,160
```

B 11

APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 73
DEFINITIONS ERRORS

4953 007400 LIST

C 11

C 11


```
4955 007400                          MOPINS  IBM,132,,,MONIBM
4958 007400                                  LIST
4959
4960
4961 007400                          DOPINS  IBM,132
5046 007400                                  LIST
```

D 11

```
5048 007400                            MOPINS  QUAD,74,T.NIL+T.SPOP
5075 007400                                    LIST
5076
5077
5078
5079
5080
5081 007400                            DOPINS  QUAD,74,T.NIL+T.SPOP
5097 007400                                    LIST
5098
5099
5100
5101
5102
5103 007400                            MOPINS  QQAD,142,T.NIL+T.SPOP
5135 007400                                    LIST
5136
5137
5138
5139
5140
5141 007400                            DOPINS  QQAD,142,T.NIL+T.SPOP
5152 007400                                    LIST
5153
5154
5155
5156
5157
5158 007400                            MOPINS  DQAD,134,T.NIL+T.SPOP
5192 007400                                    LIST
5193
5194
5195
5196 007400                            DOPINS  DQAD,134,T.NIL+T.SPOP,,D$QQAD
5199 007400                                    LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 76
DEFINITIONS     ERRORS

   5201 007400                          MOPINS  BXAV,162,T.NIL
   5219 007400                                  LIST
   5220
   5221
   5222
   5223
   5224
   5225 007400                          MOPINS  BXCT,164,T.NIL+T.SPOP
   5234 007400                                  LIST
   5235
   5236
   5237 007400                          DOPINS  BXCT,164,T.NIL+T.SPOP
   5272 007400                                  LIST
   5273
   5274
   5275
   5276
   5277
   5278 007400                          MOPINS  BXIO,166,T.NIL+T.SPOP
   5287 007400                                  LIST
   5288
   5289
   5290 007400                          DOPINS  BXIO,166,T.NIL+T.SPOP
   5302 007400                                  LIST
   5303
   5304
   5305
   5306
   5307 007400                          MOPINS  BXLC,212,T.NIL
   5340 007400                                  LIST
   5341
   5342
   5343
   5344
   5345
   5346
   5347 007400                          MOPINS  BXPP,170,T.NIL+T.SPOP
   5356 007400                                  LIST
   5357
   5358
   5359 007400                          DOPINS  BXPP,170,T.NIL+T.SPOP
   5381 007400                                  LIST
   5382
   5383
   5384
   5385
   5386
   5387
   5388 007400                          MOPINS  BXPW,172,T.NIL+T.SPOP
   5397 007400                                  LIST
   5398
   5399
   5400 007400                          DOPINS  BXPW,172,T.NIL+T.SPOP
   5417 007400                                  LIST
   5418
   5419
   5420
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 76-1
DEFINITIONS     ERRORS

   5421
   5422
   5423 007400                          MOPINS  BXRL,174,T.NIL+T.SPOP
   5432 007400                                  LIST
   5433
   5434
   5435 007400                          DOPINS  BXRL,174,T.NIL+T.SPOP
   5447 007400                                  LIST
   5448
   5449
   5450
   5451
   5452
   5453 007400                          MOPINS  BXTS,200,T.NIL
   5562 007400                                  LIST
   5563
   5564
   5565
   5566
   5567
   5568
   5569 007400                          MOPINS  BXWA,176,T.NIL
   5592 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   5594 007400                          MOPINS  BXCR,206
   5716 007400                                  LIST
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 78
DEFINITIONS     ERRORS

   5718 007400                          MOPINS  BXEX,202
   5775 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 79
DEFINITIONS     ERRORS

   5777 007400                          MOPINS  BXFX,210
   5879 007400                                  LIST
```

J 11

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 80
DEFINITIONS     ERRORS

   5881 007400                          MOPINS  BXNC,204
   5950 007400                                  LIST
```

K 11

DEFINITIONS   ERRORS

```
   5952 007400                          MOPINS  BXNL,214
   5963 007400                                  LIST
   5964
   5965
   5966 007400                          DOPINS  BXNL,214
   6104 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 82
DEFINITIONS     ERRORS

   6106 007400                          DOPINS  FGET,150,T.DIM
   6330 007400                                  LIST
   6331
   6332
   6333
   6334
   6335
   6336 007400                          DOPINS  FPUT,152,T.DIM
   6485 007400                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 83
DEFINITIONS     ERRORS

   6487 007400                          DOPINS  FPNT,154,T.DIM
   6714 007400                                  LIST
```

DEFINITIONS   ERRORS

```
   6716 007400                          MOPINS  1SCN,130
   6723 007400                                  LIST
   6724
   6725
   6726 007400                          MOPINS  SCAN,62,T.DIM
   6732 007400                                  LIST
   6733
   6734
   6735 007400                          MOPINS  1RDT,124
   6741 007400                                  LIST
   6742
   6743
   6744 007400                          MOPINS  RDCT,60,T.DIM
   7044 007400                                  LIST
```

B 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 85
DEFINITIONS     ERRORS

   7046 007400                          MOPINS  GETIDN,0
   7193 007400                                  LIST
```

C 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 86
```

C 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 86
DEFINITIONS     ERRORS

   7195 007400                          MOPINS  SATAN,0
   7404 007400                                  LIST
```

D 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 87
```

D 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 87
DEFINITIONS     ERRORS

   7406 007400                          MOPINS  OUTR,156
   7476 007400                                  LIST
```

E 12

E 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 88
DEFINITIONS     ERRORS

   7478 007400                          MOPINS  DOUTER,0
   7504 007400                                  LIST
   7505
   7506
   7507
   7508
   7509
   7510 007400                          MOPINS  PRDSET,0
   7564 007400                                  LIST
   7565
   7566
   7567
   7568
   7569
   7570 007400                          MOPINS  INSPAC,0
   7612 007400                                  LIST
```

F 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 89
```

```
7614 007400                             DOPINS  INNR,156
7615                                    .SBTTL  OPERATOR                INNER PRODUCT
7616
7617                                    ;++
7618                                    ; GENERALIZED INNER PRODUCT (MATRIX MULTIPLICATION IS A SPECIAL CASE)
7619                                    ;
7620                                    ;       This routine is called by the 'DOT'' routine during expression
7621                                    ; evaluation, so we have to do more bookkeeping than usual.
7622                                    ;
7623                                    ;       Note that the right function must be performed from the end of
7624                                    ; the arrays as in reduction, but it is not necessary that the left
7625                                    ; function be performed in that order.  Hence the elements of the
7626                                    ; result can be produced in ravel order, and DYDEXP could be used to
7627                                    ; handle integer overflow with some additional bookkeeping.
7628                                    ;
7629                                    ; TEMPORARY USAGE:
7630                                    ;
7631                                    ;       OPTMP0  - Right argument size minus first dimension reset
7632                                    ;       OPTMP1  - Right function Result type
7633                                    ;       OPTMP2  - Left function Left argument fetch routine
7634                                    ;       OPTMP3  - Left function Right argument fetch routine
7635                                    ;       OPTMP4  - Left function Result store routine
7636                                    ;       OPTMP5  - Left function number
7637                                    ;       OPTMP6  - Right argument data pointer
7638                                    ;       OPTMP7  - Combining dimension size
7639                                    ;       OPTMP8  - Left argument data pointer
7640                                    ;       OPTMP9  - Left argument size minus last dimension
7641                                    ;       OPTM10  - Right argument size minus first dimension
7642                                    ;       OPTM14  - Final result store routine
7643                                    ;       INNTMP  - Temporary for Right function result
7644                                    ;       INNTM2  - Temporary for Left function result
7645                                    ;       LARGCV  - Right function Left argument fetch routine
7646                                    ;       RARGCV  - Right function Right argument fetch routine
7647                                    ;       RESCVT  - Right function Result store routine
7648                                    ;       BLZBUB  - Right function number
7649                                    ;--
7650
7651                                    ;+
7652                                    ; FIRST OBTAIN POINTERS TO THE ARGUMENTS AND THE FUNCTION ROUTINES.
7653                                    ;-
7654
7655 000000  016503  177754                    MOV     3*SI.LEN+SI.VAL(SI),R3  ;Get pointer to right argument
7656 000004  016504  177762                    MOV     2*SI.LEN+SI.VAL(SI),R4  ;Get right function number
7657 000010                                    CALL    PRDSET                  ;Setup parameters for inner product
7658                                    ;+
7659                                    ; Check for special case of:    Left last dim = Right first dim = 1
7660                                    ;-
7661 000014                                    PUSH    <R0,R3,R4>              ;Save some registers
7662 000022  013703  003204'                   MOV     RARGBL,R3               ;Get pointer to Right argument
7663 000026  013702  003162'                   MOV     LARGBL,R2               ;Get pointer to Left argument
7664 000032  116200  000002                    MOVB    A.DIMS(R2),R0           ;Left argument's rank
7665 000036                                    CONINC  R0                      ;Cheat in case of a scalar
7666 000042  006300                            ASL     R0                      ;Convert to byte offset
7667 000044  060200                            ADD     R2,R0                   ;Add in Left pointer
7668 000046  016000  000004                    MOV     A.DIM1-2(R0),R0         ;Get last dimension of Left argument
7669 000052  020063  000006                    CMP     R0,A.DIM1(R3)           ;Does Left last dim = Right first dim?
7670 000056  001033                            BNE     1$                      ;No, Not special case
```

OPERATOR INNER PRODUCT

```
7671 000060 020027 000001                  CMP     R0,#1                   ;Is the combining dimension one?
7672 000064 001030                         BNE     1$                      ;No
7673 000066 016337 000004 003210'          MOV     A.ALEN(R3),RARGLN       ;Get number of elements in Right argument
7674 000074 016337 000004 005112'          MOV     A.ALEN(R3),OPTM10       ;Make another copy
7675 000102 016237 000004 003166'          MOV     A.ALEN(R2),LARGLN       ;Get number of elements in Left argument
7676 000110 016237 000004 005110'          MOV     A.ALEN(R2),OPTMP9       ;Copy it too
7677 000116                                CALL    INSPAC                  ;Get space for result
7678 000122                                POP     R4                      ;Get the function number back
7679 000124 022626                         CMP     (SP)+,(SP)+             ;Pop off other two registers
7680 000126 013701 003140'                 MOV     RESPTR,R1               ;Point to Result data
7681 000132 013702 003164'                 MOV     LARGPT,R2               ;Point to Left argument data
7682 000136                                CALL    DOUTER                  ;Use outer product to find result
7683 000142 000137 001274'                 JMP     INNRET                  ;Go clean up
7684
7685 000146                        1$:     POP     <R4,R3,OPTMP1>          ;Get registers back
7686 000156 010102                         MOV     R1,R2                   ;INTEMEDIATE TYPE
7687 000160 016504 177770                  MOV     SI.LEN+SI.VAL(SI),R4    ;GET LEFT FUNCTION
7688 000164 006204                         ASR     R4                      ;BYTE ACCESS
7689 000166 116400 000000G                 MOVB    DYDBTL(R4),R0           ;GET BITS
7690 000172 006304                         ASL     R4                      ;BACK TO NORMAL
7691 000174                                CALL    CMPTRT                  ;Compute Left function result type
7692 000200 010437 005100'                 MOV     R4,OPTMP5               ;Save the Left function number
7693 000204 010204                         MOV     R2,R4                   ;Copy Left fnc source intermediate type * 4
7694 000206 010203                         MOV     R2,R3                   ;Make another copy
7695 000210                                ASH     #-2,R4                  ;Get the Left fnc source intermediate type
7696 000224 060403                         ADD     R4,R3                   ;Left fnc source type index
7697 000226 016337 000000G 005072'         MOV     LARGCT(R3),OPTMP2       ;Left function left argument fetch routine
7698 000234 016337 000000G 005074'         MOV     RARGCT(R3),OPTMP3       ;Left function right argument fetch routine
7699 000242 010403                         MOV     R4,R3                   ;Copy Left fnc source intermediate type
7700 000244 063703 005070'                 ADD     OPTMP1,R3               ;Right fnc result store index
7701 000250 016337 000000G 003152'         MOV     RESCT(R3),RESCVT        ;Right function result store routine
7702 000256 060004                         ADD     R0,R4                   ;Index for Left function result store routine
7703 000260 016437 000000G 005076'         MOV     RESCT(R4),OPTMP4        ;Left function result store routine
7704 000266 060102                         ADD     R1,R2                   ;Index for Final result store routine
7705 000270 016237 000000G 005026'         MOV     RESCT(R2),OPTM14        ;Final result store routine
7706 000276 013703 003204'                 MOV     RARGBL,R3               ;POINT TO RIGHT ARG
7707 000302 013702 003162'                 MOV     LARGBL,R2               ;AND TO LEFT ARG
7708 000306 016301 000004                  MOV     A.ALEN(R3),R1           ;GET LENGTH OF RIGHT ARG
7709 000312 016200 000004                  MOV     A.ALEN(R2),R0           ;AND LENGTH OF LEFT ARG
7710 000316 005301                         DEC     R1                      ;MINUS 1 BECAUSE
7711 000320 005300                         DEC     R0                      ;WE MUST POINT TO LAST ELEMENT
7712 000322                                ASH     RARGSF,R1               ;NO, ADJUST LENGTH BY SIZE OF
7713 000336                                ASH     LARGSF,R0               ;FLOATING POINT DATA
7714 000352 060137 003206'                 ADD     R1,RARGPT               ;POINT TO END OF RIGHT ARRAY
7715 000356 060037 003164'                 ADD     R0,LARGPT               ;POINT TO END OF LEFT ARRAY
7716 000362 013737 003206' 005102'         MOV     RARGPT,OPTMP6           ;INITIALIZE RIGHT BASE POINTER
7717                                ;+
7718                                ;SET UP LEFT SIDE COUNTERS
7719                                ;-
7720 000370 005437 003160'                 NEG     LARGFL                  ;INCREMENT IS NEGATIVE
7721 000374 016201 000004                  MOV     A.ALEN(R2),R1           ;GET ARRAY LENGTH
7722 000400 116204 000002                  MOVB    A.DIMS(R2),R4           ;GET NUMBER OF DIMENSIONS
7723 000404 006304                         ASL     R4                      ;CONVERT TO BYTE COUNT
7724 000406 060204                         ADD     R2,R4                   ;MAKE POINTER ABSOLUTE
7725 000410 016404 000004                  MOV     A.DIM1-2(R4),R4         ;GET LAST DIMENSION IN LEFT ARG
7726 000414 010437 005104'                 MOV     R4,OPTMP7               ;SAVE AS LENGTH OF COMBINING DIM
7727 000420 010400                         MOV     R4,R0                   ;COPY COUNT
```

```
7728 000422                                    ASH     LARGSF,R0       ;LEFT ARG SHIFT
7729 000436  005400                            NEG     R0              ;NEGATE TO MOVE BACKWARDS
7730 000440  010037  005106'                   MOV     R0,OPTMP8       ;LEFT ARG BASE INCREMENT
7731 000444  022704  000001                    CMP     #1,R4           ;LAST DIM =1? (SPECIAL CASE: ALWAYS CONFORMS)
7732 000450  001474                            BEQ     20$             ;Yes
7733 000452  002475                            BLT     10$             ;Not null either
7734 000454  005763  000006                    TST     A.DIM1(R3)      ;Is the right dimension zero too?
7735 000460  001403                            BEQ     5$              ;Yes
7736 000462  105763  000002                    TSTB    A.DIMS(R3)      ;Is the Right argument a scalar?
7737 000466  001154                            BNE     12$             ;No, Length error
7738 000470                            5$:     PUSH    R3              ;Save pointer to right argument
7739 000472  013704  005100'                   MOV     OPTMP5,R4       ;Get Left function number
7740 000476  010203                            MOV     R2,R3           ;Copy Left argument pointer
7741 000500  062703  000006                    ADD     #A.DIM1,R3      ;Point to its first dimension
7742 000504  012701  000001                    MOV     #1,R1           ;Get multiplicative identity
7743 000510  116200  000002                    MOVB    A.DIMS(R2),R0   ;Get rank of left argument
7744 000514  005300                            DEC     R0              ;Minus one for deleted dimension
7745 000516  003407                            BLE     7$              ;Skip if zero
7746 000520                            6$:     MUL     (R3)+,R1        ;Compute size of left argument
7747 000532                                    SOB     R0,6$           ;Loop through dimension list
7748 000536  010337  005100'           7$:     MOV     R3,OPTMP5       ;Save pointer to last dimension
7749 000542  010137  005110'                   MOV     R1,OPTMP9       ;Save left argument size
7750 000546  012701  000001                    MOV     #1,R1           ;Get multiplicative identity again
7751 000552  011603                            MOV     (SP),R3         ;Get right argument pointer
7752 000554  116300  000002                    MOVB    A.DIMS(R3),R0   ;Get its rank
7753 000560  005300                            DEC     R0              ;Minus one for deleted dimension
7754 000562  003411                            BLE     9$              ;Skip if zero
7755 000564  062703  000010                    ADD     #A.DIM1+2,R3    ;Point to second dimension of right argument
7756 000570                            8$:     MUL     (R3)+,R1        ;Compute size of right argument
7757 000602                                    SOB     R0,8$           ;Loop thru dimension list
7758 000606  010137  005112'           9$:     MOV     R1,OPTM10       ;Save size of right argument
7759 000612                                    POP     R3              ;Restore pointer to right argument
7760 000614                                    MUL     OPTMP9,R1       ;Get size of result
7761 000630  001476                            BEQ     14$             ;Nothing there, Use normal method
7762 000632                                    CALL    GETIDN          ;Go get array of identity elements
7763 000636  000137  001274'                   JMP     INNRET          ;All done at last
7764
7765 000642  005037  003160'           20$:    CLR     LARGFL          ;LEFT ARG REPEATS TO MATCH RIGHT ARG
7766 000646  005000                    10$:    CLR     R0              ;CLEAR FOR DIVIDE
7767 000650                                    DIV     R4,R0           ;CALCULTE PRODUCT OF ALL BUT LAST DIM
7768 000662  010037  005110'                   MOV     R0,OPTMP9       ;TO GET LEFT ITERATION COUNT
7769                                   ;+
7770                                   ;SET UP RIGHT SIDE COUNTERS
7771                                   ;-
7772 000666  016301  000004                    MOV     A.ALEN(R3),R1   ;GET ITS SIZE
7773 000672  001007                            BNE     21$             ;It's not Null, Great
7774 000674  105762  000002                    TSTB    A.DIMS(R2)      ;Is the Left argument a scalar?
7775 000700  001673                            BEQ     5$              ;Yes
7776 000702  005737  005104'                   TST     OPTMP7          ;Is the Left argument's last dimension zero?
7777 000706  001670                            BEQ     5$              ;Yes
7778 000710  000443                            BR      12$             ;Fail all the tests, Go to LENGTH Error
7779 000712  005000                    21$:    CLR     R0              ;CLEAR FOR DIVIDE
7780 000714  016304  000006                    MOV     A.DIM1(R3),R4   ;GET FIRST DIM
7781 000720                                    DIV     R4,R0           ;COMPUTE RIGHT ARG W/O 1ST DIM
7782 000732  010037  005112'                   MOV     R0,OPTM10       ;RIGHT ARG COUNT RESET
7783 000736  010037  005066'                   MOV     R0,OPTMP0       ;   TWO OF THEM
7784 000742  020427  000001                    CMP     R4,#1           ;IS IT A SINGLE ELEMENT DIM?
```

```
7785 000746 001003                      BNE     11$                     ;NO
7786 000750 005037  003202'             CLR     RARGFL                  ;YES, DO NOTHING
7787 000754 000424                      BR      14$                     ;AND GO
7788
7789 000756                     11$:    PUSH    R4                      ;SAVE COUNT
7790 000760                             ASH     RARGSF,R0               ;SHIFT IT
7791 000774 005400                      NEG     R0                      ;NEGATE IT
7792 000776 010037  003202'             MOV     R0,RARGFL               ;IT IS THE RIGHT ARG INCREMENT
7793 001002                             POP     R0                      ;RESTORE COUNT
7794 001004 020037  005104'             CMP     R0,OPTMP7               ;CHECK FOR COMFORM
7795 001010 001406                      BEQ     14$                     ;IT DOES
7796 001012 005737  003160'             TST     LARGFL                  ;LEFT ARG SCALAR?
7797 001016 001401                      BEQ     13$                     ;YES, IS ALLRIGHT
7798
7799 001020 104412              12$:    LENERR                          ;LENGTH ERROR
7800
7801 001022 010037  005104'     13$:    MOV     R0,OPTMP7               ;NEW COUNT
7802 001026                     14$:    CALL    INSPAC                  ;Get space for Result
7803 001032 005304                      DEC     R4                      ;Origin 0 offset to Result data
7804 001034                             ASH     RESSFT,R4               ;Convert it to a byte offset
7805 001050 060437  003140'             ADD     R4,RESPTR               ;Get pointer to end of Result data
```

```
                                        ;+
                                        ;PERFOM INNER PRODUCT (FINALLY!!)
                                        ;-
001054                                  DOINNR::
001054  013700  005104'                       MOV     OPTMP7,R0               ;GET LENGTH OF COMBINING DIM
001060  013702  003164'                       MOV     LARGPT,R2               ;RESET LEFT POINER
001064  013703  003206'                       MOV     RARGPT,R3               ;RESET RIGHT POINTER
001070  012701  005016'                       MOV     #INNTM2,R1              ;POINT TO RESULT
001074                                        CALL    INNOP                   ;PERFORM FIRST (RIGHT) OP
001100  000516                               BR      INNOVR                  ;NOT READY YET

001102  005300                          1$:   DEC     R0                      ;DONE WITH THIS ROW/COLUMN?
001104  003432                               BLE     2$                      ;YES
001106  012701  003250'                       MOV     #INNTMP,R1              ;INTERMEDIATE AREA
001112                                        CALL    INNOP                   ;DO FIRST OPERATION
001116  000507                               BR      INNOVR                  ;NOT READY YET
001120                                        PUSH    <R2,R3>                 ;SAVE DATA POINTERS
001124  012701  005016'                       MOV     #INNTM2,R1              ;POINT TO RESULT
001130  012702  003250'                       MOV     #INNTMP,R2              ;TEMPORARY AREA
001134  010103                               MOV     R1,R3                   ;RESULT TO RESULT
001136                                        CALL    @OPTMP2                 ;LEFT ARG FETCH
001142                                        CALL    @OPTMP3                 ;RIGHT ARG FETCH
001146                                        POP     <R3,R2>                 ;RESTORE REGISTERS
001152  013704  005100'                       MOV     OPTMP5,R4               ;LEFT ARG OP
001156                                        CALL    @DYDTAB(R4)             ;PERFORM OP
001162                                        CALL    @OPTMP4                 ;STORE RESULT
001166  000463                               BR      INNOVR                  ;ERROR
001170  000744                               BR      1$                      ;AND LOOP ON THIS ROW/COLUMN

001172  013701  003140'              2$:      MOV     RESPTR,R1               ;Point to location in result array
001176  013737  005016'  003154'             MOV     INNTM2,RESVAL           ;Copy temperory result
001204                                        LDF     INNTM2,AC0              ;Floating form too
001214                                        CALL    @OPTM14                 ;Store the result away
001220  104402                               SYSERR                          ;Overflow should have happened before this
001222  163737  003150'  003140'             SUB     RESLEN,RESPTR           ;SAVE POINTER
001230  163737  003216'  003206'             SUB     RARGLE,RARGPT           ;BACK UP POINTER
001236  005337  005112'                       DEC     OPTM10                  ;DECREMENT COUNT
001242  003304                               BGT     DOINNR                  ;LOOP
001244  013737  005102'  003206'             MOV     OPTMP6,RARGPT           ;DONE, RESET RIGHT BASE
001252  013737  005066'  005112'             MOV     OPTMP0,OPTM10           ;RESET RIGHT COUNT
001260  063737  005106'  003164'             ADD     OPTMP8,LARGPT           ;INCREMENT LEFT BASE
001266  005337  005110'                       DEC     OPTMP9                  ;DONE ON LEFT YET?
001272  003270                               BGT     DOINNR                  ;NO, PROCESS NEXT ROW/COLUMN

                                        ;+
                                        ; Kill the argument and function frames on the SI stack and replace
                                        ; them with the Result parse frame
                                        ;-
001274  016502  177776          INNRET: MOV     SI.VAL(SI),R2   ;Save value pointer in the Result frame
001300  062705  177772                  ADD     #SI.LEN,SI      ;Skip over the Result frame
001304                                  CALL    KILL            ;Kill the Left argument frame
001310  062705  177764                  ADD     #2*SI.LEN,SI    ;Kill the function frames
001314  016504  177772                  MOV     SI.RET(SI),R4   ;Save the return address
001320                                  CALL    KILL            ;Kill the Right argument frame
001324                                  SIFRAM  R2,,R4          ;Build a new Result frame
001334                                  RETURN                  ;Return from Inner product routine
```

```
                                        ;+
                                        ;ERROR ROUTINE - FIXUP AND CONITNUE FOR 'DOINNR'  (INNER PRODUCT ROUTINE)
                                        ;-
001336  023727  003144' 000004  INNOVR: CMP     RESTYP,#.INTEGER;MUST BE INTEGER!
001344  001400                         BEQ     1$              ;It is
001346  104402                  1$:    SYSERR                  ;Not integer function
001350                                 PUSH    <R0,R2,R3,R4,R1>;SAVE REGISTERS
001362  016500  177776                 MOV     SI.VAL(SI),R0   ;POINT TO RESULT
001366  012701  000006                 MOV     #.FLOAT,R1      ;NEW STUFF IS FLOAT
001372                                 CALL    GETSM3          ;GET NEW DATA SPACE
001376  016002  000004                 MOV     A.ALEN(R0),R2   ;GET THE LENGTH
001402                                 ASH     PRECSN,R2       ;MULTIPLY FOR BYTES
001416  060201                         ADD     R2,R1           ;POINTER TO END OF NEW STUFF
001420  016002  000004                 MOV     A.ALEN(R0),R2   ;GET LENGTH AGAIN
001424  006302                         ASL     R2              ;IN BYTES, NOT WORDS
001426  067502  177770                 ADD     @SI.LEN+SI.VAL(SI),R2   ;POINTER TO DAT
001432                                 POP     R0              ;RESTORE (OLD) RESULT POINTER
001434                          4$:    LDCIF   -(R2),AC3       ;GET OLD VALUE
001442                                 STF     AC3,-(R1)       ;STORE IT
001450  020200                         CMP     R2,R0           ;STOP WHEN I FINISH
001452  001370                         BNE     4$              ;NOT YET, MY FRIEND
001454  010137  003140'                 MOV     R1,RESPTR       ;SAVE NEW POINTER
001460  012703  000006                 MOV     #.FLOAT,R3      ;RESULT IS FLOAT
001464  010337  003144'                 MOV     R3,RESTYP       ;I SAID SO...
001470  013702  005070'                 MOV     OPTMP1,R2       ;GET RESULT TYPE OF 1ST OP
001474                                 ASH     #2,R2           ;Times 4
001510  060302                         ADD     R3,R2           ;INDEX FOR STORE ROUTINE
001512  016237  000000G 003152'        MOV     RESCT(R2),RESCVT;SO LOOK IT UP
001520                                 MUL     #5.,R3          ;LOOK ALONG DIAGONAL
001534  016337  000000G 005072'        MOV     LARGCT(R3),OPTMP2 ;LEFT FETCH
001542  016337  000000G 005074'        MOV     RARGCT(R3),OPTMP3 ;RIGHT FETCH
001550  016337  000000G 005076'        MOV     RESCT(R3),OPTMP4;RESULT STORE
001556  013737  005076' 005026'        MOV     OPTMP4,OPTM14   ;Final result store routine
001564  013737  000062' 003150'        MOV     NUMLEN,RESLEN   ;RESULT LENGTH FOR LOOP
001572  013700  005100'                 MOV     OPTMP5,R0       ;GET OPERATOR
001576  002405                         BLT     5$              ;ALREADY NEGATIVE
001600  005400                         NEG     R0              ;NEGATE IT
001602  005760  000000G                TST     DYDTAB(R0)      ;ROUTINE EXISTS?
001606  001001                         BNE     5$              ;YES
001610  005400                         NEG     R0              ;NO, REVERSE IT
001612  010037  005100'          5$:    MOV     R0,OPTMP5       ;SAVE 'NEW' VALUE
001616  013737  005066' 005112'        MOV     OPTMP0,OPTM10   ;RESET COUNT
001624  014504                         MOV     -(SI),R4        ;SAVE
001626  014503                         MOV     -(SI),R3        ;  SI
001630  005745                         TST     -(SI)           ;     INFO
001632  016502  177772                 MOV     SI.RET(SI),R2   ;--RETURN INFO
001636                                 CALL    KILL            ;KILL OLD VALUE
001642  010225                         MOV     R2,(SI)+        ;NOW
001644  010325                         MOV     R3,(SI)+        ;  RESTORE
001646  010425                         MOV     R4,(SI)+        ;     THE DATA
001650                                 POP     <R4,R3,R2,R0>   ;RESTORE REGISTERS
001660  000137  001054'                 JMP     DOINNR          ;START AGAIN!


                                        ;+
                                        ;SUBROUTINE TO PERFORM FIRST (RIGHT) OPERATION
                                        ;-
```

```
7921 001664                                INNOP:  CALL    @LARGCV         ;FETCH LEFT ARG
7922 001670                                        CALL    @RARGCV         ;FETCH RIGHT ARG
7923 001674  013704  005116'                       MOV     BLZBUB,R4       ;GET OP NUMBER
7924 001700                                        CALL    @DYDTAB(R4)     ;PERFORM OP
7925 001704                                        CALL    @RESCVT         ;STORE RESULT
7926 001710  000403                                BR      1$              ;ERROR ON 'OVERFLOW'
7927 001712  062716  000002                       ADD     #2,(SP)         ;NORMAL RETURN
7928 001716                                        RETURN                  ;TRY IT, YOU'LL LIKE IT
7929                                       ;+
7930                                       ; ERROR ROUTINE FOR OVERFLOW - IF IT'S REALLY BAD, WE
7931                                       ; DO SOME OF IT AND LET THE MAIN ROUTINE ALSO TAKE SOME.
7932                                       ;-
7933 001720                                1$:     PUSH    <R0,R1>         ;Save a couple of registers
7934 001724  013700  003170'                       MOV     LARGTY,R0       ;LEFT ARG TYPE
7935 001730  016037  000000G 003176'               MOV     FLARGC(R0),LARGCV ;LEFT ARG FETCH ROUTINE
7936 001736  013700  003212'                       MOV     RARGTY,R0         ;RIGHT ARG TYPE
7937 001742  016037  000000G 003220'               MOV     FRARGC(R0),RARGCV ;RIGHT ARG FETCH ROUTINE
7938 001750  012700  000006                       MOV     #.FLOAT,R0      ;Get new result type
7939 001754  010037  005070'                       MOV     R0,OPTMP1       ;Save it
7940 001760  016037  000000G 003152'               MOV     FRESCT(R0),RESCVT ;RESULT STORE ROUTINE
7941 001766  013701  003144'                       MOV     RESTYP,R1       ;Get old result type
7942 001772  016137  000000G 005026'               MOV     FRESCT(R1),OPTM14;Get new final result store routine
7943 002000                                        ASH     #2,R1           ;Result type times 4
7944 002014  060100                                ADD     R1,R0           ;Add to left argument type
7945 002016  016037  000000G 005076'               MOV     RESCT(R0),OPTMP4;New Left function result store routine
7946 002024  013700  005116'                       MOV     BLZBUB,R0       ;GET OP NUMBER
7947 002030  003406                                BLE     2$              ;ALREADY NEGATIVE
7948 002032  005400                                NEG     R0              ;NEGATE IT
7949 002034  022760  000000G 000000G               CMP     #SYNER,DYDTAB(R0);Check to see if floating routine exists
7950 002042  001001                                BNE     2$              ;IT IS
7951 002044  005400                                NEG     R0              ;BACK TO NORMAL
7952 002046  010037  005116'          2$:     MOV     R0,BLZBUB       ;SAVE ROUTINE
7953 002052                                        POP     <R1,R0>         ;RESTORE REGISTERS
7954 002056  163702  003160'                       SUB     LARGFL,R2       ;BACK UP BOTH
7955 002062  163703  003202'                       SUB     RARGFL,R3       ;  ARGUMENTS
7956 002066  023727  003144' 000004               CMP     RESTYP,#.INTEGER;RESULT TYPE INTEGER?
7957 002074  001273                                BNE     INNOP           ;YES, TRY AGAIN
7958 002076                                        RETURN                  ;NO, ERROR RETURN
7960 002100                                        LIST
```

M 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 91
OPERATOR                INNER PRODUCT

   7962 002100                          MOPINS  SUBR,0
```

N 12

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 92
```

```
APL011  MACRO M1113  08-FEB-80 10:38  PAGE 92
OPERATOR                INNER PRODUCT

   8725 002100                                  LIST
```

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 93
OPERATOR                INNER PRODUCT

   8727 002100                          MOPINS  RAND,0
   8771 002100                                  LIST
```



```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 94
```

C 13

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 94
OPERATOR                INNER PRODUCT

      1         000001                              .END
```

D 13

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 94-1
```

APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 94-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073   C.BLOK= 000064   D.COMB= 000001   D.1ROT= 000003
AC1    =%000001          CHRLPR= 000072   C.BLSZ= 000140   D.COMP= 000001   EDICHR  005063R  025
AC2    =%000002          CHRMNS= 000137   C.BUFF= 177760   D.CONT= 000006   EDICNT  005064R  025
AC3    =%000003          CHRMUL= 000075   C.BUFL= 177762   D.CTLM= 000000   EDIFLG  005062R  025
AC4    =%000004          CHRNEG= 000100   C.CCNT= 177766   D.DEAL= 000003   EDIT    = 000003
AC5    =%000005          CHRPDL= 000271   C.CPTR= 177764   D.DECD= 000002   EIS     = 000000
ALLINT  003274R     025  CHRPLU= 000055   C.CZ  = 001000   D.DIV = 000000   ELESIZ  005132R  025
ARYRET  003276R     025  CHRPWR= 000120   C.DIG = 000040   D.DMD = 000010   ENDCOR  000004R  025
ASCQUO= 000042           CHRQST= 000121   C.FASW= 020000   D.DQAD= 177777   EOF     = 104475
A.ALEN= 000004           CHRQUO= 000113   C.FLEN= 000010   D.DROP= 000002   ERRMAX= 000166
A.ARG = 000040           CHRRBR= 000047   C.FLSH= 004000   D.ENCD= 000002   ERRNUM  004772R  025
A.ASB = 000200           CHRRPR= 000042   C.FRC = 000001   D.EQ  = 000000   ERRPTR  004770R  025
A.BITS= 000003           CHRSEM= 000074   C.IDEN= 000010   D.EXPD= 000002   ERR0    004774R  025
A.CON = 000020           CHRSMO= 000112   C.NAME= 000102   D.FGET= 000005   EXEBUF  005054R  025
A.DIMS= 000002           CHRSPA= 000040   C.NOBF= 000400   D.FMT = 000004   E$CT    005056R  025
A.DIM1= 000006           CHRUPA= 000131   C.NUM = 000020   D.FPNT= 000005   E$PT    005060R  025
A.NUM = 000006           CHR.BS= 000010   C.OP  = 000004   D.FPUT= 000005   FA.APD= 000100
A.PTR = 000000           CHR.CC= 000003   C.OPN = 010000   D.GE  = 000000   FA.CRE= 000010
A.TMP = 000010           CHR.CR= 000015   C.RDLN= 177776   D.GT  = 000000   FA.DLK= 001000
BADDIR= 104463           CHR.CZ= 000032   C.REWT= 002000   D.IBM = 000002   FA.ENB= 100000
BADERR= 104437           CHR.GT= 000046   C.RND = 000010   D.IND = 000002   FA.EXC= 002000
BLZBUB  005116R     025  CHR.LF= 000012   C.SBIT= 177756   D.INNR= 000013   FA.EXT= 000004
BRSELF  005046R     025  CHR.LT= 000043   C.SPFX= 000100   D.INTR= 000006   FA.NSP= 000100
CCNT    005072R     025  CHR.SI= 000017   C.SPGL= 000200   D.LE  = 000000   FA.NSR= 000200
CHANEX  003060R     025  CHR.SL= 000057   C.SPLX= 000001   D.LOG = 000001   FA.POS= 010000
CHAN00  000256R     025  CHR.SO= 000016   C.SPRS= 000002   D.LT  = 000000   FA.RD = 000001
CHAN01  000330R     025  CHR.TB= 000011   C.SRC = 177774   D.MAX = 000000   FA.RWD= 004000
CHAN02  000512R     025  CHR.UL= 000106   C.SRCP= 000024   D.MEM = 000002   FA.SEQ= 040000
CHAN03  000674R     025  CHR..A= 000141   C.TT  = 000004   D.MIN = 000000   FA.SHR= 000040
CHAN04  001056R     025  CHR..E= 000145   C.WIDC= 177770   D.MUL = 000000   FA.TMP= 000020
CHAN05  001240R     025  CHR..S= 000163   C.WIDR= 177772   D.NE  = 000000   FA.WCK= 020000
CHAN06  001422R     025  CHR..T= 000164   DEBUG = 000000   D.NN  = 000000   FA.WRT= 000002
CHAN07  001604R     025  CHR..U= 000165   DEFERR= 104405   D.NR  = 000000   FD.BLK= 000010
CHAN08  001766R     025  CHR..Z= 000172   DEVIL   005114R  025 D.OR  = 000000   FD.CCL= 000002
CHAN09  002150R     025  CHR..0= 000060   DEVNTF= 104470   D.PLUS= 000000   FD.COM= 020000
CHAN10  002332R     025  CHR..1= 000061   DIGITS  000006R  025 D.PSUB= 000006   FD.CR = 000002
CHAN11  002514R     025  CH.AND= 000001   DIMAX = 000020   D.PWR = 000001   FD.DIR= 000010
CHAN12  002676R     025  CMDBUF  003250R  025 DIMBLK  003112R  025 D.QQAD= 000000   FD.FTN= 000001
CHAN13  000022R     023  CMDMAX= 000010   DIMPTR  003114R  025 D.QUAD= 000000   FD.F11= 040000
CHAN14  000204R     023  CMPTRT= ****** GX DIMTYP  003116R  025 D.RES = 000000   FD.INS= 000010
CHNOPN= 104471           CNTSIZ  005130R  025 DIRFUL= 104466   D.RHO = 000000   FD.ISP= 002000
CHRAND= 000051           COMBIN  005076R  025 DOINNR  001054RG  020 D.ROT = 000003   FD.MNT= 100000
CHRBEL= 000007           COMCNT  005126R  025 DOMERR= 104417   D.SCON= 000006   FD.OSP= 004000
CHRBKR= 000133           COMERR= 104425   DOUTER= ****** GX   D.SDIF= 000006   FD.PLC= 000004
CHRBOL= 000200           COMPTR  005122R  025 DPTERR= 104416   D.SUB = 000000   FD.PRN= 000004
CHRBOX= 000114           COMRTN  003156R  025 DYDBTL= ****** GX D.SUBS= 000006   FD.PSE= 010000
CHRCMA= 000054           COMSKP  005124R  025 DYDCNT  003224R  025 D.TAKE= 000002   FD.RAH= 000001
CHRCOL= 000076           CORPTR  005120R  025 DYDTAB= ****** GX D.TRAN= 000004   FD.RAN= 000002
CHRDEL= 000107           CS.ASS= 000200   D$INNR  000000RG  020 D.UNON= 000006   FD.REC= 000001
CHRDIV= 000053           CS.BOO= 000010   D.AND = 000000   D.$MAX= 000000   FD.RWM= 000001
CHRDLR= 000176           CS.FLO= 000020   D.BXCT= 000007   D.$MIN= 000000   FD.SDI= 000020
CHRDLT= 000110           CS.IDN= 000040   D.BXIO= 000007   D.$MUL= 000000   FD.SQD= 000040
CHRDOT= 000056           CS.ONE= 000100   D.BXNL= 000007   D.$PLU= 000000   FD.TTY= 000004
CHREP = 000105           CTEPS   000152R  025 D.BXPP= 000007   D.$RES= 000000   FD.WBH= 000002
CHRERR= 104431           CTMAX   000212R  025 D.BXPW= 000007   D.$SUB= 000000   FF.CHR= 000005
CHRFIL= 000177           CTRMAX  000172R  025 D.BXRL= 000007   D.1COM= 000001   FF.NV = 000003
CHRFRO= 000135           C.ALOC= 000162   D.CIRC= 000001   D.1EXP= 000002   FF.POE= 000002
```


APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 94-2

APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 94-2
SYMBOL TABLE

```
FF.RWD= 000001          F.END  = 002000          F.VBN  = 000064          LISIZE= 000020          M.LIT  = 000205
FF.RWF= 000006          F.EOBB= 000032          F.VBSZ= 000060          LITOP   005072R 025 M.LN   = 000001
FF.SPC= 000004          F.ERR  = 000052          GETIDN= ****** GX       LNLNG  = 104541          M.MMD  = 000010
FHEADR  005012R 025     F.FACC= 000043          GETSM3= ****** GX       LOCLCT  005036R 025 M.NEG  = 000000
FHRET   005040R 025     F.FFBY= 000014          GNOME   003102R 025     LOWCOR  000002R 025 M.NOT  = 000000
FILNTF= 104467          F.FLAG= 000002          HASHC   005136R 025     LSTERP  004766R 025 M.NUB  = 000006
FIS    = 000000         F.FLB  = 000100          HASHTB  005140R 025     LWIDTH  000070R 025 M.NUM  = 000204
FISCON= 046000          F.FNAM= 000110          HSHMSK= 177601          L.BITS= 000003          M.OUTR= 000012
FIXPTR  003074R 025     F.FNB  = 000102          HSHSIZ= 000200          L.CNT  = 000002          M.PITM= 000000
FLAGS   004764R 025     F.FNDF= 000010          IASH   = ****** GX      L.PRNT= 000001          M.PRDS= 000006
FLARGC= ****** GX       F.FTYP= 000116          ICHAN   005050R 025     L.PTR  = 000000          M.QQAD= 000000
FLCBUF  003250R 025     F.FUN  = 000010          IDIV   = ****** GX      L.PTR1= 000004          M.QUAD= 000000
FLINE   005014R 025     F.FVER= 000120          ILFNAM= 104464          L.SUBS= 000002          M.RAND= 000012
FLOEXP  003270R 025     F.HEAD= 000006          ILLCHN= 104540          MACHIN= 000106          M.RAVL= 000000
FLONEG  003272R 025     F.HIBK= 000004          ILLSW  = 104565          MAXCHN= 000014          M.RCIP= 000000
FLSWTB  000222R 025     F.LBL  = 000020          IMUL   = ****** GX      MONBLK  003120R 025 M.RDCT= 000011
FLTLEN= 000010          F.LINE= 000012          INDERR= 104410          MONCVT  003132R 025 M.REVS= 000003
FNINSI  005044R 025     F.LOC  = 000004          INNOP   001664R 020     MONLEN  003130R 025 M.ROLL= 000000
FORIGN  000032R 025     F.LSTC= 000004          INNOVR  001336R 020     MONPTR  003122R 025 M.SATA= 000001
FO.APD= 000106          F.LUN  = 000042          INNRET  001274R 020     MONSFT  003126R 025 M.SCAN= 000011
FO.MFY= 000002          F.MBCT= 000054          INNTMP  003250R 025     MONTYP  003124R 025 M.SHAP= 000000
FO.RD  = 000001          F.MBC1= 000055          INNTM2  005016R 025     MONVAL  003134R 025 M.SIGN= 000000
FO.UPD= 000006          F.MBFG= 000056          INPTR   000064R 025     M.ABS  = 000000          M.SUBR= 000001
FO.WRT= 000016          F.MON  = 000002          INSNAM= 104462          M.BOL  = 000200          M.SYM  = 000203
FPMODE  000042R 025     F.NIL  = 000001          INSPAC= ****** GX       M.BXAV= 000007          M.SYN  = 000201
FPP    = 000000         F.NRBD= 000024          IOERR  = 104477          M.BXCR= 000007          M.TRAN= 000012
FPPLEN= 000004          F.NREC= 000030          I.CNT  = 177776          M.BXCT= 000007          M.$ABS= 000000
FRARGC= ****** GX       F.OPN  = 000010          I.DIM  = 177772          M.BXEX= 000007          M.$CEI= 000000
FREBLK  007400R 025     F.OVBS= 000030          I.LEN  = 000012          M.BXFX= 000007          M.$FLR= 000000
FRESCT= ****** GX       F.PREV= 000000          I.OERR= 000062          M.BXIO= 000007          M.$IDN= 000000
FUZZ    000142R 025     F.PRNT= 100000          I.OLD  = 177770          M.BXLC= 000003          M.$NEG= 000000
FXPTR1  003076R 025     F.PROT= 010000          I.PTR  = 177774          M.BXNC= 000007          M.$SIG= 000000
F$EVAL= 000100          F.RACC= 000016          I.STR  = 177766          M.BXNL= 000007          M.1RDT= 000011
F$EVI  = 000200          F.RATT= 000001          KILL   = ****** GX      M.BXPP= 000007          M.1REV= 000003
F$EXEC= 000400          F.RCNM= 000034          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
F$FUNC= 001000          F.RCTL= 000017          LABLPT  005042R 025     M.BXRL= 000007          NAMBUF  003240R 025
F.ACTL= 000076          F.READ= 004000          LARGBL  003162R 025     M.BXTS= 000007          NB.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LARGCT= ****** GX       M.BXWA= 000007          NB.DIR= 000100
F.BBFS= 000062          F.RES  = 000040          LARGCV  003176R 025     M.CEI  = 177777          NB.NAM= 000004
F.BDB  = 000070          F.RET  = 177772          LARGFL  003160R 025     M.CHR  = 000202          NB.SD1= 000400
F.BGBC= 000057          F.RSIZ= 000002          LARGLE  003174R 025     M.CMT  = 000206          NB.SD2= 001000
F.BITS= 177774          F.RTYP= 000000          LARGLN  003166R 025     M.DOUT= 000006          NB.SNM= 000040
F.BKDN= 000026          F.R2   = 000014          LARGPT  003164R 025     M.DQAD= 000006          NB.STP= 000020
F.BKDS= 000020          F.R3   = 000016          LARGSF  003172R 025     M.ENCD= 000004          NB.SVR= 000010
F.BKEF= 000050          F.R4   = 000020          LARGTY  003170R 025     M.EXE  = 000000          NB.TYP= 000002
F.BKP1= 000051          F.SEQN= 000100          LARGVA  003200R 025     M.EXP  = 000001          NB.VER= 000001
F.BKST= 000024          F.SPDV= 000072          LBINC   005100R 025     M.FACT= 000001          ND.DEV= 000200
F.BKVB= 000064          F.SPUN= 000074          LBUFF   003316R 025     M.FIL  = 000177          ND.EXT= 000002
F.BR   = 000020          F.STBK= 000036          LBUFLN= 000621          M.FLR  = 177777          ND.NAM= 000004
F.CHR  = 000075          F.STK  = 000022          LCOUNT  005106R 025     M.FMT  = 000004          ND.PPN= 000100
F.CNTG= 000034          F.STOP= 000004          LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.CODE= 000010          F.STP  = 000001          LENTAB  000054R 025     M.GDWN= 000003          ND.$E$= 000020
F.DFNB= 000046          F.SUSF= 000002          LEXPTR  003100R 025     M.GETI= 000006          ND.$NM= 000000
F.DSPT= 000044          F.TRC  = 000002          LFTOP   005066R 025     M.IBM  = 000003          ND.$N$= 000040
F.DVNM= 000134          F.UNIT= 000136          LINC    005102R 025     M.IDN  = 000000          ND.$PN= 001400
F.EFBK= 000010          F.URBD= 000020          LINE    005016R 025     M.INSP= 000006          NEGSGN  005063R 025
F.EFN  = 000050          F.VAL  = 177776          LININC  005026R 025     M.IOTA= 000000          NINPT1  003302R 025
```


APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 94-3

APLC11 MACRO M1113 08-FEB-80 10:38 PAGE 94-3
SYMBOL TABLE

```
NINPT2  003300R     025 OPTMP7  005104R     025 ROVER   003312R     025 SYNER = ****** GX       T.FUN = 010000
NINSTK  003304R     025 OPTMP8  005106R     025 RSTS  = 000000          SYNERR= 104407         T.LIST= 000400
NINTMP  003310R     025 OPTMP9  005110R     025 RSX   = 000115          SYSERR= 104402         T.LIT = 000020
NINTYP  003306R     025 OPTM10  005112R     025 RTOP    005070R     025 SYSTAT  000000R     024 T.MON = 000010
NOTAVL= 104472          OPTM14  005026R     025 RT11  = 000000          SY.BAD= 000001         T.NIL = 000004
NOTOPN= 104473          ORIGIN  000030R     025 R.BITS= 000003          SY.BRK= 100000         T.NUM = 002000
NPRT1   004762R     025 OUTPTR  000066R     025 R.FIX = 000001          SY.BS = 000010         T.OPFN= 000001
NULL    007364R     025 OVROPR= 000013          R.LCT = 000002          SY.CLR= 000040         T.OTHR= 040000
NUMCHN= 000020          PATCH   007344R     025 R.LN  = 000002          SY.ERP= 000020         T.SOP = 000002
NUMDFL  003230R     025 PRDSET= ****** GX       R.LNUM= 000010          SY.NOU= 000100         T.SPOP= 000040
NUMEFL  003233R     025 PRECSN  000052R     025 R.NEXT= 000006          SY.OPN= 000002         T.TMP = 000200
NUMESN  003236R     025 RANDAC  003064R     025 R.OFF = 000000          SY.PMT= 001000         T.VAL = 100000
NUMFLG  003230R     025 RANDY   003062R     025 R.PREV= 000004          SY.SRT= 000400         T.VAR = 001000
NUMLEN  000062R     025 RANERR= 104411          R.PTR = 000000          SY.SVC= 000004         UPPER = 177400
NUMNEG  003232R     025 RARGBL  003204R     025 R.SEQ = 000003          SY.SVL= 000200         UTNAME  000344R     023
NUMQFL  003234R     025 RARGCT= ****** GX       R.VAR = 000002          S.AB  = 177776         VALERR= 104413
NUMRIG  003231R     025 RARGCV  003220R     025 R6    =%000006          S.ASB = 100000         VARBLE  007400R     025
NUMWID  003226R     025 RARGFL  003202R     025 R7    =%000007          S.BFHD= 000020         VARBLS  000000R     025
NUMZFL  003235R     025 RARGLE  003216R     025 SI    =%000005          S.BITS= 000002         VERNUM  000000R     025
NYIERR= 104432          RARGLN  003210R     025 SIDAMN= 104433          S.CHR1= 000003         VRSION= 000002
N.BLSZ= 000036          RARGPT  003206R     025 SIFRAM  003314R     025 S.FATT= 000016         WIDTH   000072R     025
N.DESC= 000020          RARGSF  003214R     025 SISTAK  006350R     025 S.FDB = 000140         WROUND  000020R     025
N.DEV = 000032          RARGTY  003212R     025 SITOP   006346R     025 S.FLB = 040000         WSDATE  000136R     025
N.DEVN= 000034          RARGVA  003222R     025 SITP    007144R     025 S.FNAM= 000006         WSFULL= 104403
N.DID = 000024          RBASE   005074R     025 SIWARN= 000200          S.FNB = 000036         WSNAME  000074R     025
N.DVNM= 000032          RCOUNR  005112R     025 SIZTAB  000044R     025 S.FNBW= 000017         WSTIME  000132R     025
N.EXT = 000014          RCOUNT  005110R     025 SI.BIT= 177774          S.FNTY= 000004         $FIS  = ****** G
N.FID = 000000          REDCNT  005070R     025 SI.LEN= 177772          S.FTYP= 000002         $FPP  = ****** G
N.FNAM= 000006          REDIGN  005066R     025 SI.RET= 177772          S.LEN = 000002         $FPPLN= ****** G
N.FTYP= 000014          REDOFF  005072R     025 SI.VAL= 177776          S.LINK= 177774         $LDCIF= ****** GX
N.FVER= 000016          REENTR  003060R     025 SKPSIZ  005134R     025 S.NFEN= 000020         $LDF  = ****** GX
N.NAME= 000006          REGTYP= 000001          SPHOLD  000004R     024 S.TYPE= 140000         $MACHN= ****** G
N.NEXT= 000022          RESBIT  003142R     025 SPINIT  000002R     024 S.VAL = 000000         $RSTS = ****** G
N.PPN = 000024          RESCT = ****** GX       STKINC= 000774          TKDIMS  005072R     025 $RSX  = ****** G
N.STAT= 000020          RESCVT  003152R     025 STOP  = 104420          TKFRAM  005066R     025 $RT11 = ****** G
N.UNIT= 000034          RESERR= 104440          SUBCNT  003104R     025 TKLEFT  005070R     025 $STF  = ****** GX
OBUFLN= 001000          RESLEN  003150R     025 SUBDCT  003106R     025 TMODE   000403R     023 .BOOLE= 000002
OCHAN   005052R     025 RESPTR  003140R     025 SUBPTR  003110R     025 TMPTAG= 000166          .CHARA= 000000
OMODE   000402R     023 RESSFT  003146R     025 SYMB1   005344R     025 TTYBUF  004140R     025 .DEF. = 000002
OPTMP0  005066R     025 RESTYP  003144R     025 SYMF    005340R     025 TTYLEN= 000622          .FLOAT= 000006
OPTMP1  005070R     025 RESULT  003136R     025 SYMPT1  005346R     025 TURKEY= 104436          .FSRCB= ****** G
OPTMP2  005072R     025 RESVAL  003154R     025 SYMTAB  005352R     025 TYPE  = 000166          .INTEG= 000004
OPTMP3  005074R     025 RINC    005104R     025 SYMTBF= 104404          T.BOL = 020000         .MOLUN  000404R     023
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMTOP  005350R     025 T.CON = 004000         ...GBL= 000000
OPTMP5  005100R     025 ROTINC  005070R     025 SYMTP   006272R     025 T.DIM = 000100         ...TPC= 000020
OPTMP6  005102R     025 ROUND   000010R     025 SYMWRN= 000054

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
```


APLO11 MACRO M1113 08-FEB-80 10:38 PAGE 94-4

G 13

```
APLO11  MACRO M1113  08-FEB-80 10:38  PAGE 94-4
SYMBOL TABLE

VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    002100      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13336 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:03:18
APLO11,DM1:[305,301]APLO11/-SP=APLRX1,APLDEF,APLO11,APLOP,END
```

H 13

```
**FILE**ID**APRINT
```

H 13

FILEID**APRINT

```
  AAAAAA      PPPPPPPP      RRRRRRRR       IIIIII     NN      NN   TTTTTTTTTT
  AAAAAA      PPPPPPPP      RRRRRRRR       IIIIII     NN      NN   TTTTTTTTTT
AA      AA    PP      PP    RR      RR       II       NN      NN       TT
AA      AA    PP      PP    RR      RR       II       NN      NN       TT
AA      AA    PP      PP    RR      RR       II       NNNN    NN       TT
AA      AA    PP      PP    RR      RR       II       NNNN    NN       TT
AA      AA    PPPPPPPP      RRRRRRRR         II       NN  NN  NN       TT
AA      AA    PPPPPPPP      RRRRRRRR         II       NN  NN  NN       TT
AAAAAAAAAA    PP            RR   RR          II       NN    NNNN       TT
AAAAAAAAAA    PP            RR   RR          II       NN    NNNN       TT
AA      AA    PP            RR     RR        II       NN      NN       TT          ....
AA      AA    PP            RR     RR        II       NN      NN       TT          ....
AA      AA    PP            RR      RR    IIIIII     NN      NN       TT          ....
AA      AA    PP            RR      RR    IIIIII     NN      NN       TT          ....

LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

I 13

.MAIN. MACRO M1113 08-FEB-80 10:42

.MAIN. MACRO M1113 08-FEB-80 10:42
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    APRINT           PRINT AN ARRAY
```


APRINT MACRO M1113 08-FEB-80 10:42 PAGE 1

APRINT MACRO M1113 08-FEB-80 10:42 PAGE 1

```
                                ;
                                ;  PRINT UTILITY FOR RSX-11/M
                                ;
                                ;  BUILD COMMANDS:
                                ;
                                ;  MAC APRINT=APRINT
                                ;  TKB APRINT=APRINT
                                ;  INSTALL APRINT
                                ;
                                ;  RUN COMMAND:
                                ;
                                ;  APRINT
                                ;
                                ;  THIS PROGRAM WILL PRINT ANY ASCII FILE ON THE LINE PRINTER.
                                ;  IT WILL ACCEPT WILDCARD SPECIFICATIONS AND WILL PRINT ALL
                                ;  FILES THAT MATCH THE WILDCARD SPECIFICATION.
                                ;
                                ;  THE PROGRAM WILL PRINT A HEADER ON EACH PAGE CONTAINING
                                ;  THE FILE NAME, TYPE, VERSION, AND PAGE NUMBER.
                                ;
                                ;  THE PROGRAM WILL EXPAND TABS TO EVERY 8TH COLUMN.
                                ;
                                ;  THE PROGRAM WILL PRINT 60 LINES PER PAGE.
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING SWITCHES:
                                ;
                                ;  /NOHEADER  - DO NOT PRINT HEADER ON EACH PAGE
                                ;  /NOFORMFEED - DO NOT PRINT FORM FEED AT END OF FILE
                                ;  /NOTRUNCATE - DO NOT TRUNCATE LINES LONGER THAN 132 CHARACTERS
                                ;  /NOEXPAND  - DO NOT EXPAND TABS
                                ;  /NOPAGE    - DO NOT PRINT PAGE NUMBER IN HEADER
                                ;  /NOVERSION - DO NOT PRINT VERSION NUMBER IN HEADER
                                ;  /NOFILE    - DO NOT PRINT FILE NAME IN HEADER
                                ;  /NOTYPE    - DO NOT PRINT FILE TYPE IN HEADER
                                ;  /NOLOG     - DO NOT PRINT LOGGING MESSAGES
                                ;  /NOWARN    - DO NOT PRINT WARNING MESSAGES
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING COMMANDS:
                                ;
                                ;  /HELP      - PRINT THIS HELP FILE
                                ;  /EXIT      - EXIT THE PROGRAM
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING OPTIONS:
                                ;
                                ;  /LOG       - PRINT LOGGING MESSAGES
                                ;  /WARN      - PRINT WARNING MESSAGES
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING PARAMETERS:
                                ;
                                ;  /LP:N      - PRINT ON LP:N
                                ;  /COPIES:N  - PRINT N COPIES
                                ;  /QUEUE:N   - QUEUE THE JOB TO PRINT N COPIES
                                ;  /DELETE    - DELETE THE FILE AFTER PRINTING
                                ;  /FORMS:N   - USE FORMS TYPE N
                                ;  /LIMIT:N   - PRINT N LINES PER PAGE
                                ;  /WIDTH:N   - PRINT N CHARACTERS PER LINE
                                ;  /TABS:N    - EXPAND TABS TO EVERY NTH COLUMN
                                ;  /TITLE:"STRING" - PRINT "STRING" AS THE TITLE
                                ;  /SUBTITLE:"STRING" - PRINT "STRING" AS THE SUBTITLE
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING QUALIFIERS:
                                ;
                                ;  /BEFORE:DD-MMM-YY - PRINT FILES CREATED BEFORE DD-MMM-YY
                                ;  /SINCE:DD-MMM-YY - PRINT FILES CREATED SINCE DD-MMM-YY
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING FILE SPECIFICATIONS:
                                ;
                                ;  DEV:FILNAM.TYP;VER
                                ;
                                ;  WILDCARDS ARE ALLOWED IN ALL FIELDS.
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULTS:
                                ;
                                ;  /LP:LP0:
                                ;  /COPIES:1
                                ;  /QUEUE:1
                                ;  /FORMS:0
                                ;  /LIMIT:60
                                ;  /WIDTH:132
                                ;  /TABS:8
                                ;  /TITLE:""
                                ;  /SUBTITLE:""
                                ;  /BEFORE:01-JAN-77
                                ;  /SINCE:01-JAN-77
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULT FILE SPECIFICATIONS:
                                ;
                                ;  DEV:*.*;*
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULT SWITCHES:
                                ;
                                ;  /HEADER
                                ;  /FORMFEED
                                ;  /TRUNCATE
                                ;  /EXPAND
                                ;  /PAGE
                                ;  /VERSION
                                ;  /FILE
                                ;  /TYPE
                                ;  /NOLOG
                                ;  /WARN
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULT OPTIONS:
                                ;
                                ;  /NOLOG
                                ;  /WARN
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULT PARAMETERS:
                                ;
                                ;  /LP:LP0:
                                ;  /COPIES:1
                                ;  /QUEUE:1
                                ;  /FORMS:0
                                ;  /LIMIT:60
                                ;  /WIDTH:132
                                ;  /TABS:8
                                ;  /TITLE:""
                                ;  /SUBTITLE:""
                                ;  /BEFORE:01-JAN-77
                                ;  /SINCE:01-JAN-77
                                ;
                                ;  THE PROGRAM WILL ACCEPT THE FOLLOWING DEFAULT QUALIFIERS:
                                ;
```

K 13

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

L 13

```
APL     MACRO M1113  08-FEB-80 10:42  PAGE 42
```

```
APL     MACRO M1113  08-FEB-80 10:42  PAGE 42
DEFINITIONS     ERRORS

  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 44
```

```
                                .TITLE  APRINT
                                .IDENT  \V02.03\
                                
                                .SBTTL  APRINT          PRINT AN ARRAY
                                
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'APRINT' prints an array in the standard APL format-
                                ;       an extra blank line for each different dimension.
                                ;       (understand?)  All 'APRINT' does is set up parameters
                                ;       for the 'NPRINT' routine.  'NPRINT' is a beautiful
                                ;       little recursive routine which actually does the
                                ;       printing by calling 'FLOUT' or 'OUTCHR' for each element in the
                                ;       array, and by inserting blank lines when necessary.
                                ;
                                ; REGISTER ASSIGNMENTS:
                                ;       R0      - Scratch
                                ;       R1      - Current level in recursion (current dimensio)
                                ;       R2      - Pointer to element to print next
                                ;       R3      - Pointer to array dimension list
                                ;       R4      - Flag for literal/numeric array printing
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     #<Array to be printed>,R0
                                ;       CALL    APRINT
                                ;
                                ; INPUTS:
                                ;       R0      - Array to be printed.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       The array is output on the current output channel.
                                ;
                                ; ROUTINES CALLED:
                                ;       NPRINT  - The recursive print routine
                                ;--
000000                          .PSECT  CODE
```

```
                                                        .ENABL  LSB
                                                APRNT1::
000000                                                  PUSH    <R4,R3>         ;Save a couple of registers
000000                                                  CMP     #OP.FMT,R4      ;Is the function Format?
000004  022704  000000G                                BEQ     1$              ;Yes
000010  001403                                          CMP     #OP.ENC,R4      ;Is the function Encode?
000012  022704  000000G                                BNE     2$              ;No
000016  001005                                          MOV     #10,R3          ;Setup for alternate NUMOUT entry points
000020  012703  000010                          1$:     BR      3$              ;And go
000024  000403
                                                APRINT::
000026                                                  PUSH    <R4,R3>         ;Save a couple of registers
000026                                          2$:     CLR     R3              ;Normal NUMOUT entry points
000032  005003                                  3$:     PUSH    <R2,R1,ALLINT>  ;SAVE REGISTERS, OUTPUT PACKING
000034                                                  TST     A.ALEN(R0)      ;NULL ARRAY?
000044  005760  000004                                 BEQ     8$              ;YES, EXIT
000050  001506                                          MOV     SI,ALLINT       ;FUDGE FOR SEMI-CLOSE PACKING
000052  010537  003274'                                MOVB    A.BITS(R0),R4   ;GET TYPE OF SOURCE
000056  116004  000003                                 BIC     #^C<A.NUM>,R4   ;CLEAR GARBAGE BITS
000062  042704  177771                                 BEQ     4$              ;CHARACTER SPECIAL CASE
000066  001415                                          PUSH    R4              ;Save the data type
000070                                                  ADD     R4,R3           ;Normal or alternate Numout entry point
000072  060403                                         MOV     OUTCT(R3),RESCVT;Output routine
000074  016337  000000' 003152'                        ASL     R4              ;TIMES 2
000102  006304                                         ASL     R4              ;TIMES 4
000104  006304                                         ADD     (SP)+,R4        ;TIMES 5
000106  062604                                         MOV     MONCT(R4),MONCVT;FETCH ROUTINE
000110  016437  000000G 003132'                        MOV     #NPRT0,R4       ;NO, GET OFFSET FOR LITERAL HANDLING
000116  012704  000010                          4$:     MOV     R0,R3           ;COPY POINTER TO ARRAY
000122  010003                                         ADD     #A.DIM1,R3      ;POINT TO DIMENSION LIST
000124  062703  000006                                 MOV     @R0,R2          ;COPY POINTER AGAIN
000130  011002                                         MOVB    A.DIMS(R0),R1   ;GET # OF DIMS (LEVEL)
000132  116001  000002                                 CONINC  R1              ;CHEAT FOR SCALAR
000136                                                  CMPB    #DIMAX,R1       ;TOO MANY?
000142  122701  000020                                 BHIS    5$              ;NO
000146  103001                                         SYSERR                  ;TOO MANY DIMS!
000150  104402                                  5$:     ADD     R1,R3           ;BYTES NOT WORDS?!!
000152  060103                                         ADD     R1,R3           ;POINT TO END OF DIMENSION LIST
000154  060103                                         MOV     R2,NPRT1        ;SAVE INITIAL POINTER
000156  010237  004762'                                BIS     #SY.NOU,SYSTAT  ;KILL ALL OUTPUT
000162  052737  000100  000000'                        MOV     #1,NUMWID       ;ASSUME SMALL NUMBERS
000170  012737  000001  003226'                        CLRB    NUMEFL          ;MARK NO EXPONENT FOUND
000176  105037  003233'                                TST     @SP             ;CLOSE PACKING?
000202  005716                                         BNE     7$              ;YES, DON'T PRE-TREAT
000204  001023                                         TST     R4              ;CHARACTER MODE?
000206  005704                                         BEQ     6$              ;YES, EXPAND IT IF NECESSARY
000210  001417                                         CMP     R1,#1           ;VECTOR OR SCALAR ?
000212  020127  000001                                 BLE     7$              ;YES, DON'T PRE-SCAN
000216  003416                                         PUSH    <R0,R1,R2,R3,R4>;SAVE ALL REGS
000220                                                  CALL    NPRINT          ;FIGURE OUT NUMERIC FIELD WIDTH
000232                                                  POP     <R4,R3,R2,R1,R0>;RESTORE ALL REGS
000236                                          6$:     CLR     ALLINT          ;NO, FANCY FORMATTING
000250  005037  003274'                          7$:     BIC     #SY.NOU,SYSTAT  ;ALLOW OUTPUT AGAIN
000254  042737  000100  000000'                        CALL    NPRINT          ;PRINT ARRAY
000262                                          8$:     POP     <ALLINT,R1,R2,R3,R4>;RESTORE REGISTERS
000266                                                  RETURN                  ;AND EXIT
000302                                                  .DSABL  LSB
```

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 45-1
APRINT          PRINT AN ARRAY

     98
```



```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 46
```

```
                                .PSECT  CODE

                                NPRINT::
                                        CALL    BRKCHK          ;CHECK TO SEE IF HE WANT'S TO STOP
                                        TST     R1              ;LEVEL =0?
                                        BNE     8$              ;NO, KEEP RECURSING
                                        PUSH    R0              ;SAVE R0
                                        MOV     #1,R0           ;ASSUME CHR OUTPUT
                                        TST     R4              ;CHR OUTPUT?
                                        BEQ     1$              ;YES
                                        MOV     NUMWID,R0       ;NUMBER WIDTH
                                        ADD     #3,R0           ;PLUS SIGN, DECIMAL POINT, TRAILING SPACE
                                        TSTB    NUMEFL          ;EXPONENTS BEING PRINTED?
                                        BEQ     1$              ;NO
                                        ADD     #4,R0           ;YES, SAVE ROOM FOR THEM
                                1$:     PUSH    R0              ;SAVE COUNT
                                        MOV     OCHAN,R0        ;POINT TO OUTPUT CHANNEL
                                        CMP     (SP)+,C.WIDC(R0);ENOUGH ROOM ON LINE?
                                        BLOS    5$              ;YES!
                                        TST     C.WIDR(R0)      ;Line width in force?
                                        BEQ     5$              ;Nope, ignore it
                                        PUSH    R3              ;SAVE R3
                                        CALL    NEWLIN          ;No, skip a line
                                        MOV     #6,R3           ;SPACE IN 6
                                        TST     ALLINT          ;SHORT STUFF ?
                                        BNE     2$              ;YES, SHORT...
                                        TST     R4              ;CHARACTER?
                                        BEQ     3$              ;YES - 6 SPACES
                                        MOV     NUMWID,R3       ;GET NUMBER FIELD WIDTH
                                        ADD     #2,R3           ;PLUS SPACE AND SIGN
                                        TSTB    NUMEFL          ;EXPONENT BEING PRINTED?
                                        BEQ     3$              ;NO
                                        ADD     #4,R3           ;YES, SAVE ROOM FOR IT

                                2$:     TST     R4              ;NUMERIC?
                                        BEQ     4$              ;NO, NO SPACES
                                3$:     MOV     #CHRSPA,R0      ;PRINT SPACES
                                        CALL    OUTCHR          ;TO MARK CONTINUED
                                        SOB     R3,3$           ;LINE (LONG VECTOR)
                                4$:     POP     R3              ;RESTORE R3

                                5$:     POP     R0              ;RESTORE R0
                                        ADD     R4,PC           ;BRANCH TO ARRAY TYPE ROUTINE

                                ; PRINT LITERAL ARRAY
                                6$:     MOVB    (R2)+,R0        ;GET NEXT CHAR
                                        CALL    OUTCHR          ;PRINT IT
                                        BR      10$             ;AND RETURN FROM THIS LEVEL

                                ;PRINT NUMERIC ARRAY
                                7$:     CALL    @MONCVT         ;ELSE GET NEXT ELEMENT IN ARRAY
                                        CALL    @RESCVT         ;PRINT THE NUMBER
                                        CMP     2(SP),#1        ;LAST IN LINE?
                                        BLE     10$             ;YES, NO SPACE
                                        PUSH    R0              ;SAVE R0
                                        MOVB    #CHRSPA,R0      ;GET A SPACE
                                        CALL    OUTCHR          ;PRINT IT
```

```
    157 000530                                  POP     R0              ;RESTORE R0
    158 000532  000420                          BR      10$             ;AND RETURN FROM THIS LEVEL
    159
    160         000010          NPRT0=7$-6$                             ;OFFSET FOR PRINTING LITERAL ARRAYS
    161
    162 000534  010300          8$:     MOV     R3,R0                   ;GET DIMS POINTER
    163 000536  160100                  SUB     R1,R0                   ;POINT TO DIM FOR THIS LEVEL
    164 000540  160100                  SUB     R1,R0                   ;COUNT IS IN WORDS NOT BYTES!
    165 000542  011000                  MOV     @R0,R0                  ;GET THE DIM COUNT
    166 000544                          PUSH    R0                      ;SAVE COUNT FOR THIS LEVEL
    167 000546  020237  004762'         CMP     R2,NPRT1                ;INITIAL RECURSION?
    168 000552  001402                  BEQ     9$                      ;YES, NO LINES SKIPPED THEN
    169 000554                          CALL    NEWLIN                  ;Else skip a line for the new level
    170 000560  005301          9$:     DEC     R1                      ;BUMP DOWN ONE LEVEL
    171 000562                          CALL    NPRINT                  ;RECURSE TO PRINT THIS LEVEL
    172 000566  005316                  DEC     @SP                     ;COUNT RECURSION ON THIS LEVEL
    173 000570  003373                  BGT     9$                      ;NOT DONE, KEEP PRINTING
    174 000572                          POP     R0                      ;CLEAN UP STACK
    175 000574  005201          10$:    INC     R1                      ;INCREASE LEVEL COUNT TO RESET COUNT
    176 000576                          RETURN                          ;RETURN FROM THIS LEVEL
```

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 47
APRINT          PRINT AN ARRAY

    178 000000                                      .PSECT  TABLES
    179                                             .EVEN
    180
    181 000000                              OUTCT::
    182 000000  000000G 000000G 000000G             .WORD   SYSER,  BOLOUT, INTOUT, FLOUT
        000006  000000G
    183 000010  000000G 000000G 000000G             .WORD   SYSER,  BOLOUT, INTOU1, FLOU1
        000016  000000G
```



```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 48
```

F 14

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 48
APRINT          PRINT AN ARRAY

      1         000001                          .END
```

G 14

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 48-1
SYMBOL TABLE
```

APRINT MACRO M1113 08-FEB-80 10:42 PAGE 48-1
SYMBOL TABLE

```
AC0    =%000000          CHREP = 000105      CTMAX   000212R 025 D.CONT= 000006      EDICNT  005064R 025
AC1    =%000001          CHRERR= 104431      CTRMAX  000172R 025 D.CTLM= 000000      EDIFLG  005062R 025
AC2    =%000002          CHRFIL= 000177      C.ALOC= 000162      D.DEAL= 000003      EDIT   = 000003
AC3    =%000003          CHRFRO= 000135      C.BLOK= 000064      D.DECD= 000002      EIS    = 000000
AC4    =%000004          CHRLBR= 000073      C.BLSZ= 000140      D.DIV = 000000      ELESIZ  005132R 025
AC5    =%000005          CHRLPR= 000072      C.BUFF= 177760      D.DMD = 000010      ENDCOR  000004R 025
ALLINT  003274R     025  CHRMNS= 000137      C.BUFL= 177762      D.DQAD= 177777      EOF    = 104475
APRINT  000026RG    020  CHRMUL= 000075      C.CCNT= 177766      D.DROP= 000002      ERRMAX= 000166
APRNT1  000000RG    020  CHRNEG= 000100      C.CPTR= 177764      D.ENCD= 000002      ERRNUM  004772R 025
ARYRET  003276R     025  CHRPDL= 000271      C.CZ   = 001000     D.EQ   = 000000     ERRPTR  004770R 025
ASCQUO= 000042           CHRPLU= 000055      C.DIG = 000040      D.EXPD= 000002      ERR0    004774R 025
A.ALEN= 000004           CHRPWR= 000120      C.FASW= 020000      D.FGET= 000005      EXEBUF  005054R 025
A.APG = 000040           CHRQST= 000121      C.FLEN= 000010      D.FMT = 000004      E$CT    005056R 025
A.ASB = 000200           CHRQUO= 000113      C.FLSH= 004000      D.FPNT= 000005      E$PT    005060R 025
A.BITS= 000003           CHRRBR= 000047      C.FRC = 000001      D.FPUT= 000005      FA.APD= 000100
A.CON = 000020           CHRRPR= 000042      C.IDEN= 000010      D.GE   = 000000     FA.CRE= 000010
A.DIMS= 000002           CHRSEM= 000074      C.NAME= 000102      D.GT   = 000000     FA.DLK= 001000
A.DIM1= 000006           CHRSMO= 000112      C.NOBF= 000400      D.IBM = 000002      FA.ENB= 100000
A.NUM = 000006           CHRSPA= 000040      C.NUM = 000020      D.IND = 000002      FA.EXC= 002000
A.PTR = 000000           CHRUPA= 000131      C.OP   = 000004     D.INNR= 000013      FA.EXT= 000004
A.TMP = 000010           CHR.BS= 000010      C.OPN = 010000      D.INTR= 000006      FA.NSP= 000100
BADDIR= 104463           CHR.CC= 000003      C.RDLN= 177776      D.LE   = 000000     FA.NSR= 000200
BADERR= 104437           CHR.CR= 000015      C.REWI= 002000      D.LOG = 000001      FA.POS= 010000
BLZBUB  005116R     025  CHR.CZ= 000032      C.RND = 000010      D.LT   = 000000     FA.RD  = 000001
BOLOUT= ****** GX        CHR.GT= 000046      C.SBIT= 177756      D.MAX = 000000      FA.RWD= 004000
BRKCHK= ****** GX        CHR.LF= 000012      C.SPFX= 000100      D.MEM = 000002      FA.SEQ= 040000
BRSELF  005046R     025  CHR.LT= 000043      C.SPGL= 000200      D.MIN = 000000      FA.SHR= 000040
CCNT    005072R     025  CHR.SI= 000017      C.SPLX= 000001      D.MUL = 000000      FA.TMP= 000020
CHANEX  003060R     025  CHR.SL= 000057      C.SPRS= 000002      D.NE   = 000000     FA.WCK= 020000
CHAN00  000256R     025  CHR.SO= 000016      C.SRC = 177774      D.NN   = 000000     FA.WRT= 000002
CHAN01  000330R     025  CHR.TB= 000011      C.SRCP= 000024      D.NR   = 000000     FD.BLK= 000010
CHAN02  000512R     025  CHR.UL= 000106      C.TT   = 000004     D.OR   = 000000     FD.CCL= 000002
CHAN03  000674R     025  CHR..A= 000141      C.WIDC= 177770      D.PLUS= 000000      FD.COM= 020000
CHAN04  001056R     025  CHR..E= 000145      C.WIDR= 177772      D.PSUB= 000006      FD.CR  = 000002
CHAN05  001240R     025  CHR..S= 000163      DEBUG  = 000000     D.PWR  = 000001     FD.DIR= 000010
CHAN06  001422R     025  CHR..T= 000164      DEFERR= 104405      D.QQAD= 000000      FD.FTN= 000001
CHAN07  001604R     025  CHR..U= 000165      DEVIL   005114R 025 D.QUAD= 000000      FD.F11= 040000
CHAN08  001766R     025  CHR..Z= 000172      DEVNTF= 104470      D.RES  = 000000     FD.INS= 000010
CHAN09  002150R     025  CHR..0= 000060      DIGITS  000006R 025 D.RHO  = 000000     FD.ISP= 002000
CHAN10  002332R     025  CHR..1= 000061      DIMAX  = 000020     D.ROT  = 000003     FD.MNT= 100000
CHAN11  002514R     025  CH.AND= 000001      DIMBLK  003112R 025 D.SCON= 000006      FD.OSP= 004000
CHAN12  002676R     025  CMDBUF  003250R 025 DIMPTR  003114R 025 D.SDIF= 000006      FD.PLC= 000004
CHAN13  000022R     023  CMDMAX= 000010      DIMTYP  003116R 025 D.SUB  = 000000     FD.PRN= 000004
CHAN14  000204R     023  CNTSIZ  005130R 025 DIRFUL= 104466      D.SUBS= 000006      FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R 025 DOMERR= 104417      D.TAKE= 000002      FD.RAH= 000001
CHRAND= 000051           COMCNT  005126R 025 DPTERR= 104416      D.TRAN= 000004      FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425      DYDCNT  003224R 025 D.UNON= 000006      FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R 025 D.AND  = 000000     D.$MAX= 000000      FD.RWM= 000001
CHRBOL= 000200           COMRTN  003156R 025 D.BXCT= 000007      D.$MIN= 000000      FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R 025 D.BXIO= 000007      D.$MUL= 000000      FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R 025 D.BXNL= 000007      D.$PLU= 000000      FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200      D.BXPP= 000007      D.$RES= 000000      FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010      D.BXPW= 000007      D.$SUB= 000000      FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020      D.BXRL= 000007      D.1COM= 000001      FF.NV  = 000003
CHRDLR= 000176           CS.IDN= 000040      D.CIRC= 000001      D.1EXP= 000002      FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100      D.COMB= 000001      D.1ROT= 000003      FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R 025 D.COMP= 000001      EDICHR  005063R 025 FF.RWF= 000006
```


APRINT MACRO M1113 08-FEB-80 10:42 PAGE 48-2

SYMBOL TABLE

```
FF.SPC= 000004          F.FACC= 000043  HASHC   005136R 025 L.SUBS= 000002      M.QUAD= 000000
FHEADR  005012R 025     F.FFBY= 000014  HASHTB  005140R 025 MACHIN= 000106      M.RAND= 000012
FHRET   005040R 025     F.FLAG= 000002  HSHMSK= 177601      MAXCHN= 000014      M.RAVL= 000000
FILNTF= 104467          F.FLB = 000100  HSHSIZ= 000200      MONBLK  003120R 025 M.RCIP= 000000
FIS   = 000000          F.FNAM= 000110  ICHAN   005050R 025 MONCT = ****** GX   M.RDCT= 000011
FISCON= 046000          F.FNB = 000102  ILFNAM= 104464      MONCVT  003132R 025 M.REVS= 000003
FIXPTR  003074R 025     F.FNDF= 000010  ILLCHN= 104540      MONLEN  003130R 025 M.ROLL= 000000
FLAGS   004764R 025     F.FTYP= 000116  ILLSW = 104565      MONPTR  003122R 025 M.SATA= 000001
FLCBUF  003250R 025     F.FUN = 000010  INDERR= 104410      MONSFT  003126R 025 M.SCAN= 000011
FLINE   005014R 025     F.FVER= 000120  INNTMP  003250R 025 MONTYP  003124R 025 M.SHAP= 000000
FLOEXP  003270R 025     F.HEAD= 000006  INNTM2  005016R 025 MONVAL  003134R 025 M.SIGN= 000000
FLONEG  003272R 025     F.HIBK= 000004  INPTR   000064R 025 M.ABS = 000000      M.SUBR= 000001
FLOUT = ****** GX       F.LBL = 000020  INSNAM= 104462      M.BOL = 000200      M.SYM = 000203
FLOU1 = ****** GX       F.LINE= 000012  INTOUT= ****** GX   M.BXAV= 000007      M.SYN = 000201
FLSWTB  000222R 025     F.LOC = 000004  INTOU1= ****** GX   M.BXCR= 000007      M.TRAN= 000012
FLTLEN= 000010          F.LSTC= 000004  IOERR = 104477      M.BXCT= 000007      M.$ABS= 000000
FNINSI  005044R 025     F.LUN = 000042  I.CNT = 177776      M.BXEX= 000007      M.$CEI= 000000
FORIGN  000032R 025     F.MBCT= 000054  I.DIM = 177772      M.BXFX= 000007      M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055  I.LEN = 000012      M.BXIO= 000007      M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056  I.OERR= 000062      M.BXLC= 000003      M.$NEG= 000000
FO.RD = 000001          F.MON = 000002  I.OLD = 177770      M.BXNC= 000007      M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001  I.PTR = 177774      M.BXNL= 000007      M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024  I.STR = 177766      M.BXPP= 000007      M.1REV= 000003
FPMODE  000042R 025     F.NREC= 000030  LABERR= 104406      M.BXPW= 000007      M.1SCN= 000011
FPP   = 000000          F.OPN = 000010  LABLPT  005042R 025 M.BXRL= 000007      NAMBUF  003240R     025
FPPLEN= 000004          F.OVBS= 000030  LARGBL  003162R 025 M.BXTS= 000007      NB.DEV= 000200
FREBLK  007400R 025     F.PREV= 000000  LARGCV  003176R 025 M.BXWA= 000007      NB.DIR= 000100
FUZZ    000142R 025     F.PRNT= 100000  LARGFL  003160R 025 M.CEI = 177777      NB.NAM= 000004
FXPTR1  003076R 025     F.PROT= 010000  LARGLE  003174R 025 M.CHR = 000202      NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016  LARGLN  003166R 025 M.CMT = 000206      NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001  LARGPT  003164R 025 M.DOUT= 000006      NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034  LARGSF  003172R 025 M.DQAD= 000006      NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017  LARGTY  003170R 025 M.ENCD= 000004      NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000  LARGVA  003200R 025 M.EXE = 000000      NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040  LBINC   005100R 025 M.EXP = 000001      NB.VER= 000001
F.BBFS= 000062          F.RES = 000040  LBUFF   003316R 025 M.FACT= 000001      ND.DEV= 000200
F.BDB = 000070          F.RET = 177772  LBUFLN= 000621      M.FIL = 000177      ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002  LCOUNT  005106R 025 M.FLR = 177777      ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000  LENERR= 104412      M.FMT = 000004      ND.PPN= 000100
F.BKDN= 000026          F.R2  = 000014  LENTAB  000054R 025 M.GDUP= 000003      ND.$EX= 000000
F.BKDS= 000020          F.R3  = 000016  LEXPTR  003100R 025 M.GDWN= 000003      ND.$ES= 000020
F.BKEF= 000050          F.R4  = 000020  LFTOP   005066R 025 M.GETI= 000006      ND.$NM= 000000
F.BKP1= 000051          F.SEQN= 000100  LINC    005102R 025 M.IBM = 000003      ND.$NS= 000040
F.BKST= 000024          F.SPDV= 000072  LINE    005016R 025 M.IDN = 000000      ND.$PN= 001400
F.BKVB= 000064          F.SPUN= 000074  LININC  005026R 025 M.INSP= 000006      NEGSGN  005063R     025
F.BR  = 000020          F.STBK= 000036  LISIZE= 000020      M.IOTA= 000000      NEWLIN= ****** GX
F.CHR = 000075          F.STK = 000022  LITOP   005072R 025 M.LIT = 000205      NINPT1  003302R     025
F.CNTG= 000034          F.STOP= 000004  LNLNG = 104541      M.LN  = 000001      NINPT2  003300R     025
F.CODE= 000010          F.STP = 000001  LOCLCT  005036R 025 M.MMD = 000010      NINSTK  003304R     025
F.DFNB= 000046          F.SUSP= 000002  LOWCOR  000002R 025 M.NEG = 000000      NINTMP  003310R     025
F.DSPT= 000044          F.TRC = 000002  LSTERP  004766R 025 M.NOT = 000000      NINTYP  003306R     025
F.DVNM= 000134          F.UNIT= 000136  LWIDTH  000070R 025 M.NUB = 000006      NOTAVL= 104472
F.EFBK= 000010          F.URBD= 000020  L.BITS= 000003      M.NUM = 000204      NOTOPN= 104473
F.EFN = 000050          F.VAL = 177776  L.CNT = 000002      M.OUTR= 000012      NPRINT  000304RG    020
F.END = 002000          F.VBN = 000064  L.PRNT= 000001      M.PITM= 000000      NPRTO = 000010
F.EOBB= 000032          F.VBSZ= 000060  L.PTR = 000000      M.PRDS= 000006      NPRT1   004762R     025
F.ERR = 000052          GNOME   003102R 025 L.PTR1= 000004  M.QQAD= 000000      NULL    007364R     025
```


SYMBOL TABLE

APRINT MACRO M1113 08-FEB-80 10:42 PAGE 48-3
SYMBOL TABLE

```
NUMCHN= 000020          OP.FMT= ****** GX       RT11  = 000000          SYSER = ****** GX       T.DIM = 000100
NUMDFL  003230R   025   ORIGIN  000030R   025  R.BITS= 000003          SYSERR= 104402          T.FUN = 010000
NUMEFL  003233R   025   OUTCHR= ****** GX       R.FIX = 000001          SYSTAT  000000R  024  T.LIST= 000400
NUMESN  003236R   025   OUTCT   000000RG  002  R.LCT = 000002          SY.BAD= 000001          T.LIT = 000020
NUMFLG  003230R   025   OUTPTR  000066R   025  R.LN  = 000002          SY.BRK= 100000          T.MON = 000010
NUMLEN  000062R   025   OVROPR= 000000          R.LNUM= 000010          SY.BS = 000010          T.NIL = 000004
NUMNEG  003232R   025   PATCH   007344R   025  R.NEXT= 000006          SY.CLR= 000040          T.NUM = 002000
NUMQFL  003234R   025   PRECSN  000052R   025  R.OFF = 000000          SY.ERP= 000020          T.OPFN= 000001
NUMRIG  003231R   025   RANDAC  003064R   025  R.PREV= 000004          SY.NOU= 000100          T.OTHR= 040000
NUMWID  003226R   025   RANDY   003062R   025  R.PTR = 000000          SY.OPN= 000002          T.SOP = 000002
NUMZFL  003235R   025   RANERR= 104411          R.SEQ = 000003          SY.PMT= 001000          T.SPOP= 000040
NYIERR= 104432          RARGBL  003204R   025  R.VAR = 000002          SY.SRT= 000400          T.TMP = 000200
N.BLSZ= 000036          RARGCV  003220R   025  R6    =%000006          SY.SVC= 000004          T.VAL = 100000
N.DESC= 000020          RARGFL  003202R   025  R7    =%000007          SY.SVL= 000200          T.VAR = 001000
N.DEV = 000032          RARGLE  003216R   025  SI    =%000005          S.AB  = 177776          UPPER = 177400
N.DEVN= 000034          RARGLN  003210R   025  SIDAMN= 104433          S.ASB = 100000          UTNAME  000344R   023
N.DID = 000024          RARGPT  003206R   025  SIFRAM  003314R  025  S.BFHD= 000020          VALERR= 104413
N.DVNM= 000032          RARGSF  003214R   025  SISTAK  006350R  025  S.BITS= 000002          VARBLE  007400R   025
N.EXT = 000014          RARGTY  003212R   025  SITOP   006346R  025  S.CHR1= 000003          VARBLS  000000R   025
N.FID = 000000          RARGVA  003222R   025  SITP    007144R  025  S.FATT= 000016          VERNUM  000000R   025
N.FNAM= 000006          RBASE   005074R   025  SIWARN= 000200          S.FDB = 000140          VRSION= 000002
N.FTYP= 000014          RCOUNR  005112R   025  SIZTAB  000044R  025  S.FLB = 040000          WIDTH   000072R   025
N.FVER= 000016          RCOUNT  005110R   025  SI.BIT= 177774          S.FNAM= 000006          WROUND  000020R   025
N.NAME= 000006          REDCNT  005070R   025  SI.LEN= 177772          S.FNB = 000036          WSDATE  000136R   025
N.NEXT= 000022          REDIGN  005066R   025  SI.RET= 177772          S.FNBW= 000017          WSFULL= 104403
N.PPN = 000024          REDOFF  005072R   025  SI.VAL= 177776          S.FNTY= 000004          WSNAME  000074R   025
N.STAT= 000020          REENTR  003060R   025  SKPSIZ  005134R  025  S.FTYP= 000002          WSTIME  000132R   025
N.UNIT= 000034          RESBIT  003142R   025  SPHOLD  000004R  024  S.LEN = 000002          $FIS  = ****** G
OBUFLN= 001000          RESCVT  003152R   025  SPINIT  000002R  024  S.LINK= 177774          $FPP  = ****** G
OCHAN   005052R   025   RESERR= 104440          STKINC= 000774          S.NFEN= 000020          $FPPLN= ****** G
OMODE   000402R   023   RESLEN  003150R   025  STOP  = 104420          S.TYPE= 140000          $MACHN= ****** G
OPTMP0  005066R   025   RESPTR  003140R   025  SUBCNT  003104R  025  S.VAL = 000000          $RSTS = ****** G
OPTMP1  005070R   025   RESSFT  003146R   025  SUBDCT  003106R  025  TKDIMS  005072R   025  $RSX  = ****** G
OPTMP2  005072R   025   RESTYP  003144R   025  SUBPTR  003110R  025  TKFRAM  005066R   025  $RT11 = ****** G
OPTMP3  005074R   025   RESULT  003136R   025  SYMB1   005344R  025  TKLEFT  005070R   025  .BOOLE= 000002
OPTMP4  005076R   025   RESVAL  003154R   025  SYMF    005340R  025  TMODE   000403R   023  .CHARA= 000000
OPTMP5  005100R   025   RINC    005104R   025  SYMPT1  005346R  025  TMPTAG= 000166          .DEF. = 000002
OPTMP6  005102R   025   RLTBAS  005066R   025  SYMTAB  005352R  025  TTYBUF  004140R   025  .FLOAT= 000006
OPTMP7  005104R   025   ROTINC  005070R   025  SYMTBF= 104404          TTYLEN= 000622          .FSRCB= ****** G
OPTMP8  005106R   025   ROUND   000010R   025  SYMTOP  005350R  025  TURKEY= 104436          .INTEG= 000004
OPTMP9  005110R   025   ROVER   003312R   025  SYMTP   006272R  025  TYPE  = 000166          .MOLUN  000404R   023
OPTM10  005112R   025   RSTS  = 000000          SYMWRN= 000054          T.BOL = 020000          ...GBL= 000000
OPTM14  005026R   025   RSX   = 000115          SYNERR= 104407          T.CON = 004000          ...TPC= 000020
OP.ENC= ****** GX       RTOP    005070R   025

. ABS.  000000  000
        000000  001
TABLES  000020  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
```


APRINT MACRO M1113 08-FEB-80 10:42 PAGE 48-4

J 14

```
APRINT  MACRO M1113  08-FEB-80 10:42  PAGE 48-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000600      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13064 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:57
APRINT,DM1:[305,301]APRINT/-SP=APLRX1,APLDEF,APRINT,END
```

K 14

```
**FILE**ID**ARYSET
```

```
**FILE**ID**ARYSET
```

```
  AAAAAA      RRRRRRRR      YY       YY     SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT
  AAAAAA      RRRRRRRR      YY       YY     SSSSSSSS   EEEEEEEEEE   TTTTTTTTTT
AA      AA    RR      RR    YY       YY    SS         EE               TT
AA      AA    RR      RR    YY       YY    SS         EE               TT
AA      AA    RR      RR      YY   YY      SS         EE               TT
AA      AA    RR      RR      YY   YY      SS         EE               TT
AA      AA    RRRRRRRR          YY          SSSSSS     EEEEEEEE         TT
AA      AA    RRRRRRRR          YY            SSSSSS   EEEEEEEE         TT
AAAAAAAAAA    RR   RR           YY                 SS  EE               TT
AAAAAAAAAA    RR   RR           YY                 SS  EE               TT
AA      AA    RR     RR         YY                 SS  EE               TT          ....
AA      AA    RR     RR         YY                 SS  EE               TT          ....
AA      AA    RR       RR       YY         SSSSSSSS    EEEEEEEEEE       TT          ....
AA      AA    RR       RR       YY         SSSSSSSS    EEEEEEEEEE       TT          ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```



```
.MAIN.  MACRO M1113  08-FEB-80 10:44
TABLE OF CONTENTS
```

.MAIN. MACRO M1113 08-FEB-80 10:44
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    ARYSET           SET UP ARRAY SUBSCRIPT POINTERS
```


ARYSET MACRO M1113 08-FEB-80 10:44 PAGE 1

ARYSET MACRO M1113 08-FEB-80 10:44 PAGE 1

```
                                RT11    =  0            ;NOT RT-11
                                RSTS    =  0            ;NOT RSTS/E
                                RSX     = 'M            ;RSX-11/M

                                EIS     =  0            ;NO EIS
                                FIS     =  0            ;NO FIS
                                FPP     =  0            ;NO FPP

                                FPPLEN  =  4.           ;SINGLE PRECISION

                                DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```


ARYSET MACRO M1113 08-FEB-80 10:44 PAGE 2

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 10:44  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 44
```

```
                        .TITLE  ARYSET
                        .IDENT  \V02.03\

                        .SBTTL  ARYSET          SET UP ARRAY SUBSCRIPT POINTERS

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;         This routine sets up a list of index frames
                        ;       on the sp stack and initializes the pointers for
                        ;       the 'NXTIND' routine.  These index frames are used
                        ;       for subscripting arrays by the 'VAL' and 'ASSIGN' routines.
                        ;
                        ; CALLING SEQUENCE:
                        ;       CALL    ARYSET
                        ;
                        ; INPUTS:
                        ;       R0      - Points into SI stack at frame containg the
                        ;                 array to be subscripted. This frame must
                        ;                 follow a list frame containing the subscript
                        ;                 list
                        ;
                        ; IMPLICIT INPUTS:
                        ;       NONE
                        ;
                        ; OUTPUTS:
                        ;       R2      - Points to 1st element in array to subscript
                        ;       R3      - Contains level count (# of dimensions in array)
                        ;       R4      - Points to top of index frame list
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;       All index frames are initialize for the 'NXTIND' routine.
                        ;
                        ; ROUTINES CALLED:
                        ;       NONE
                        ;
                        ; SIDE EFFECTS:
                        ;       NONE
                        ;--

000000                  .PSECT  CODE
```

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 45
ARYSET           SET UP ARRAY SUBSCRIPT POINTERS

     42 000000                                  ARYSET::
     43 000000  012637  003276'                        MOV     (SP)+,ARYRET    ;SAVE RETURN
     44 000004                                         PUSH    <R2,R3,R4>      ;SAVE REGISTERS
     45 000012  010004                                 MOV     R0,R4           ;SAVE ARRAY POINTER
     46 000014  014000                                 MOV     -(R0),R0        ;POINT TO ARG
     47 000016  011001                                 MOV     @R0,R1          ;POINT TO FIRST ELEMENT
     48 000020  116037  000003  003306'                MOVB    A.BITS(R0),NINTYP;GET BITS
     49 000026  042737  177771  003306'                BIC     #^C<A.NUM>,NINTYP;CLEAR GARBAGE
     50 000034  010137  003310'                        MOV     R1,NINTMP       ;SAVE ARRAY VALUE PTR
     51 000040  116003  000002                        MOVB    A.DIMS(R0),R3   ;GET DIM COUNT
     52 000044  001403                                 BEQ     1$              ;CAN'T SUBSCRIPT SCALARS
     53 000046  022703  000020                        CMP     #DIMAX,R3       ;TOO MANY DIMS?
     54 000052  003001                                 BGT     2$              ;NO, GOOD
     55 000054  104410                        1$:     INDERR                  ;YES, INDEX ERROR
     56 000056  062700  000006                2$:     ADD     #A.DIM1,R0      ;POINT TO FIRST DIM
     57 000062  010037  003300'                        MOV     R0,NINPT2       ;SAVE DIM LIST POINTER
     58 000066  062704  177772                         ADD     #SI.LEN,R4      ;POINT TO SUBSCRIPT LIST
     59 000072  014401                                 MOV     -(R4),R1        ;POINT TO LIST BLOCK
     60 000074  126103  000002                        CMPB    L.CNT(R1),R3    ;CORRECT NUMBER OF INDICES??
     61 000100  001365                                 BNE     1$              ;NOPE, INDEX ERROR
     62 000102  011100                                 MOV     @R1,R0          ;POINT TO LIST POINTER LIST
     63 000104  060300                                 ADD     R3,R0           ;ADD # OF PTRS
     64 000106  060300                                 ADD     R3,R0           ;WORDS NOT BYTES
     65 000110  010037  003302'                        MOV     R0,NINPT1       ;AND SAVE PTR TO CALCULATE POINERS
     66 000114  010300                                 MOV     R3,R0           ;COPY DIM COUNT
     67 000116  001756                                 BEQ     1$              ;NEVER HAPPENS?
     68 000120  013701  003300'                        MOV     NINPT2,R1       ;POINT TO FIRST DIM
     69 000124  006300                                 ASL     R0              ;MULTIPLY # OF DIMS BY 2
     70 000126  060037  003300'                        ADD     R0,NINPT2       ;POINT PAST END OF DIM LIST
     71 000132  010300                                 MOV     R3,R0           ;GET BACK # OF DIMS
     72 000134  010637  003304'                        MOV     SP,NINSTK       ;SAVE POINTER TO KILL INDEX FRAMES
     73 000140  013704  003302'                        MOV     NINPT1,R4       ;POINT TO LIST OF POINTERS
```

```
                                        ;SET UP INDEX FRAMES ON STACK

000144  005046                 3$:    CLR     -(SP)              ;CLEAR COUNT
000146  005046                        CLR     -(SP)              ;CLEAR ARRAY POINTER
000150                                 PUSH    (R1)+              ;SET MAX DIM
000152  005046                        CLR     -(SP)              ;CLEAR OLD VALUE
000154  005046                        CLR     -(SP)              ;CLEAR STRIDE
000156  014402                        MOV     -(R4),R2           ;POINT TO NEXT SUBSC ARG
000160  001405                        BEQ     7$                 ;SKIP IF INDEX OMITTED
000162  132762  000006  000003       BITB    #A.NUM,A.BITS(R2)  ;INDEX NUMERIC??
000170  001001                        BNE     7$                 ;YES, IT MUST BE
000172  104417                        DOMERR                     ;LITERAL INDEX ARG ILLEGAL
000174                           7$:    SOB     R0,3$              ;AND SET FRAME FOR EACH DIM

000200  010604                        MOV     SP,R4              ;COPY POINTER PAST TOP FRAME
000202  062704  000012                ADD     #I.LEN,R4          ;POINT TO REF PT OF TOP FRAME
000206  013702  003310'               MOV     NINTMP,R2          ;GET BACK PTR TO ARRAY VALUES
000212                                 PUSH    <R2,R3,R4>         ;SAVE INITIALIZATION REGISTERS
000220  012701  000001                MOV     #1,R1              ;GET INITIAL STRIDE
000224  010300                        MOV     R3,R0              ;GET DIM COUNT
                                        ;+
                                        ; CALCULATE STRIDE VECTOR AND PLACE IN INDEX FRAMES
                                        ;-
000226  010164  177766           4$:    MOV     R1,I.STR(R4)       ;FILL IN STRIDE
000232  005300                        DEC     R0                 ;DONE?
000234  001412                        BEQ     5$                 ;YES
000236                                 MUL     I.DIM(R4),R1       ;NO, MULTIPLY STRIDE BY MAX DIM
000252  103700                        BCS     1$                 ;CHECK FOR DIM TOO BIG
000254  062704  000012                ADD     #I.LEN,R4          ;BACK UP ONE FRAME
000260  000762                        BR      4$                 ;AND PROCESS NEXT FRAME

000262  011604                   5$:    MOV     @SP,R4             ;GET BACK INITIAL STACK POINTER
                                        ;+
                                        ; INITIALIZE VALUES IN INDEX FRAMES BY CALLING
                                        ; 'NXTIND' ROUTINE WHICH EXITS IMMEDIATELY AFTER
                                        ; INITIALIZING VALUES
                                        ;-
000264  012703  000001                MOV     #1,R3              ;FAKE COUNT TO INITIALIZE FRAMES
000270                                 CALL    NXTIND             ;SET UP POINTER
000274  102401                        BVS     6$                 ;DONE (WE HAD BETTER BE!)
000276  104402                        SYSERR                     ;NO, INCORRECT INITIALIZATION
000300                           6$:    POP     <R4,R3,R2>         ;GET BACK INITIALIZATION REGS
000306  013704  003304'               MOV     NINSTK,R4          ;START ON INNERMOST INDEX FRAME
000312  000177  003276'               JMP     @ARYRET            ;AND RETURN
```

F 15

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 48
ARYSET           SET UP ARRAY SUBSCRIPT POINTERS

      1          000001                              .END
```

G 15

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 48-1
SYMBOL TABLE
```

ARYSET MACRO M1113 08-FEB-80 10:44 PAGE 48-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135  C.BLOK= 000064  D.DECD= 000002  EIS    = 000000
AC1    =%000001          CHRLBR= 000073  C.BLSZ= 000140  D.DIV = 000000  ELESIZ 005132R 025
AC2    =%000002          CHRLPR= 000072  C.BUFF= 177760  D.DMD = 000010  ENDCOR 000004R 025
AC3    =%000003          CHRMNS= 000137  C.BUFL= 177762  D.DQAD= 177777  EOF    = 104475
AC4    =%000004          CHRMUL= 000075  C.CCNT= 177766  D.DROP= 000002  ERRMAX= 000166
AC5    =%000005          CHRNEG= 000100  C.CPTR= 177764  D.ENCD= 000002  ERRNUM 004772R 025
ALLINT  003274R 025      CHRPDL= 000271  C.CZ  = 001000  D.EQ  = 000000  ERRPTR 004770R 025
ARYRET  003276R 025      CHRPLU= 000055  C.DIG = 000040  D.EXPD= 000002  ERRO   004774R 025
ARYSET  000000RG 020     CHRPWR= 000120  C.FASW= 020000  D.FGET= 000005  EXEBUF C05054R 025
ASCQUO= 000042           CHRQST= 000121  C.FLEN= 000010  D.FMT = 000004  E$CT   005056R 025
A.ALEN= 000004           CHRQUO= 000113  C.FLSH= 004000  D.FPNT= 000005  E$PT   005060R 025
A.ARG = 000040           CHRRBR= 000047  C.FRC = 000001  D.FPUT= 000005  FA.APD= 000100
A.ASB = 000200           CHRRPR= 000042  C.IDEN= 000010  D.GE  = 000000  FA.CRE= 000010
A.BITS= 000003           CHRSEM= 000074  C.NAME= 000102  D.GT  = 000000  FA.DLK= 001000
A.CON = 000020           CHRSMO= 000112  C.NOBF= 000400  D.IBM = 000002  FA.ENB= 100000
A.DIMS= 000002           CHRSPA= 000040  C.NUM = 000020  D.IND = 000002  FA.EXC= 002000
A.DIM1= 000006           CHRUPA= 000131  C.OP  = 000004  D.INNR= 000013  FA.EXT= 000004
A.NUM = 000006           CHR.BS= 000010  C.OPN = 010000  D.INTR= 000006  FA.NSP= 000100
A.PTR = 000000           CHR.CC= 000003  C.RDLN= 177776  D.LE  = 000000  FA.NSR= 000200
A.TMP = 000010           CHR.CR= 000015  C.REWT= 002000  D.LOG = 000001  FA.POS= 010000
BADDIR= 104463           CHR.CZ= 000032  C.RND = 000010  D.LT  = 000000  FA.RD = 000001
BADERR= 104437           CHR.GT= 000046  C.SBIT= 177756  D.MAX = 000000  FA.RWD= 004000
BLZBUB  005116R 025      CHR.LF= 000012  C.SPFX= 000100  D.MEM = 000002  FA.SEQ= 040000
BRSELF  005046R 025      CHR.LT= 000043  C.SPGL= 000200  D.MIN = 000000  FA.SHR= 000040
CCNT    005072R 025      CHR.SI= 000017  C.SPLX= 000001  D.MUL = 000000  FA.TMP= 000020
CHANEX  003060R 025      CHR.SL= 000057  C.SPRS= 000002  D.NE  = 000000  FA.WCK= 020000
CHAN00  000256R 025      CHR.SO= 000016  C.SRC = 177774  D.NN  = 000000  FA.WRT= 000002
CHAN01  000330R 025      CHR.TB= 000011  C.SRCP= 000024  D.NR  = 000000  FD.BLK= 000010
CHAN02  000512R 025      CHR.UL= 000106  C.TT  = 000004  D.OR  = 000000  FD.CCL= 000002
CHAN03  000674R 025      CHR..A= 000141  C.WIDC= 177770  D.PLUS= 000000  FD.COM= 020000
CHAN04  001056R 025      CHR..E= 000145  C.WIDR= 177772  D.PSUB= 000006  FD.CR = 000002
CHAN05  001240R 025      CHR..S= 000163  DEBUG = 000000  D.PWR = 000001  FD.DIR= 000010
CHAN06  001422R 025      CHR..T= 000164  DEFERR= 104405  D.QQAD= 000000  FD.FTN= 000001
CHAN07  001604R 025      CHR..U= 000165  DEVIL   005114R 025 D.QUAD= 000000  FD.F11= 040000
CHAN08  001766R 025      CHR..Z= 000172  DEVNTF= 104470  D.RES = 000000  FD.INS= 000010
CHAN09  002150R 025      CHR..0= 000060  DIGITS  000006R 025 D.RHO = 000000  FD.ISP= 002000
CHAN10  002332R 025      CHR..1= 000061  DIMAX = 000020  D.ROT = 000003  FD.MNT= 100000
CHAN11  002514R 025      CH.AND= 000001  DIMBLK  003112R 025 D.SCON= 000006  FD.OSP= 004000
CHAN12  002676R 025      CMDBUF  003250R 025 DIMPTR  003114R 025 D.SDIF= 000006  FD.PLC= 000004
CHAN13  000022R 023      CMDMAX= 000010  DIMTYP  003116R 025 D.SUB = 000000  FD.PRN= 000004
CHAN14  000204R 023      CNTSIZ  005130R 025 DIRFUL= 104466  D.SUBS= 000006  FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R 025 DOMERR= 104417  D.TAKE= 000002  FD.RAH= 000001
CHRAND= 000051           COMCNT  005126R 025 DPTERR= 104416  D.TRAN= 000004  FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425  DYDCNT  003224R 025 D.UNON= 000006  FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R 025 D.AND = 000000  D.$MAX= 000000  FD.RWM= 000001
CHRBOL= 000200           COMRTN  003156R 025 D.BXCT= 000007  D.$MIN= 000000  FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R 025 D.BXIO= 000007  D.$MUL= 000000  FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R 025 D.BXNL= 000007  D.$PLU= 000000  FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200      D.BXPP= 000007  D.$RES= 000000  FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010      D.BXPW= 000007  D.$SUB= 000000  FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020      D.BXRL= 000007  D.1COM= 000001  FF.NV = 000003
CHRDLR= 000176           CS.IDN= 000040      D.CIRC= 000001  D.1EXP= 000002  FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100      D.COMB= 000001  D.1ROT= 000003  FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R 025 D.COMP= 000001  EDICHR 005063R 025 FF.RWF= 000006
CHREP = 000105           CTMAX   000212R 025 D.CONT= 000006  EDICNT 005064R 025 FF.SPC= 000004
CHRERR= 104431           CTRMAX  000172R 025 D.CTLM= 000000  EDIFLG 005062R 025 FHEADR 005012R 025
CHRFIL= 000177           C.ALOC= 000162      D.DEAL= 000003  EDIT  = 000003      FHRET  005040R 025
```


ARYSET MACRO M1113 08-FEB-80 10:44 PAGE 48-2
SYMBOL TABLE

SYMBOL TABLE

```
FILNTF= 104467          F.FNB  = 000102          ILFNAM= 104464          MONPTR  003122R 025 M.SATA= 000001
FIS    = 000000          F.FNDF= 000010          ILLCHN= 104540          MONSFT  003126R 025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565          MONTYP  003124R 025 M.SHAP= 000000
FIXPTR  003074R 025 F.FUN = 000010          IMUL   = ****** GX       MONVAL  003134R 025 M.SIGN= 000000
FLAGS   004764R 025 F.FVER= 000120          INDERR= 104410          M.ABS  = 000000         M.SUBR= 000001
FLCBUF  003250R 025 F.HEAD= 000006          INNTMP  003250R 025 M.BOL  = 000200         M.SYM  = 000203
FLINE   005014R 025 F.HIBK= 000004          INNTM2  005016R 025 M.BXAV= 000007         M.SYN  = 000201
FLOEXP  003270R 025 F.LBL  = 000020          INPTR   000064R 025 M.BXCR= 000007         M.TRAN= 000012
FLONEG  003272R 025 F.LINE= 000012          INSNAM= 104462          M.BXCT= 000007         M.$ABS= 000000
FLSWTB  000222R 025 F.LOC  = 000004          IOERR  = 104477          M.BXEX= 000007         M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 0000C4          I.CNT  = 177776          M.BXFX= 000007         M.$FLR= 000000
FNINSI  005044R 025 F.LUN  = 000042          I.DIM  = 177772          M.BXIO= 000007         M.$IDN= 000000
FORIGN  000032R 025 F.MBCT= 000054          I.LEN  = 000012          M.BXLC= 000003         M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062          M.BXNC= 000007         M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD  = 177770          M.BXNL= 000007         M.1RDT= 000011
FO.RD  = 000001          F.MON  = 000002          I.PTR  = 177774          M.BXPP= 000007         M.1REV= 000003
FO.UPD= 000006          F.NIL  = 000001          I.STR  = 177766          M.BXPW= 000007         M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406          M.BXRL= 000007         NAMBUF  003240R 025
FPMODE  000042R 025 F.NREC= 000030          LABLPT  005042R 025 M.BXTS= 000007         NB.DEV= 000200
FPP    = 000000          F.OPN  = 000010          LARGBL  003162R 025 M.BXWA= 000007         NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R 025 M.CEI  = 177777         NB.NAM= 000004
FREBLK  007400R 025 F.PREV= 000000          LARGFL  003160R 025 M.CHR  = 000202         NB.SD1= 000400
FUZZ    000142R 025 F.PRNT= 100000          LARGLE  003174R 025 M.CMT  = 000206         NB.SD2= 001000
FXPTR1  003076R 025 F.PROT= 010000          LARGLN  003166R 025 M.DOUT= 000006         NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R 025 M.DQAD= 000006         NB.STP= 000020
F$EVI  = 000200          F.RATT= 000001          LARGSF  003172R 025 M.ENCD= 000004         NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R 025 M.EXE  = 000000         NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R 025 M.EXP  = 000001         NB.VER= 000001
F.ACTL= 000076          F.READ= 004000          LBINC   005100R 025 M.FACT= 000001         ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R 025 M.FIL  = 000177         ND.EXT= 000002
F.BBFS= 000062          F.RES  = 000040          LBUFLN= 000621          M.FLR  = 177777         ND.NAM= 000004
F.BDB  = 000070          F.RET  = 177772          LCOUNT  005106R 025 M.FMT  = 000004         ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002          LENERR= 104412          M.GDUP= 000003         ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R 025 M.GDWN= 000003         ND.$ES= 000020
F.BKDN= 000026          F.R2   = 000014          LEXPTR  003100R 025 M.GETI= 000006         ND.$NM= 000000
F.BKDS= 000020          F.R3   = 000016          LFTOP   005066R 025 M.IBM  = 000003         ND.$NS= 000040
F.BKEF= 000050          F.R4   = 000020          LINC    005102R 025 M.IDN  = 000000         ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R 025 M.INSP= 000006         NEGSGN  005063R 025
F.BKST= 000024          F.SPDV= 000072          LININC  005026R 025 M.IOTA= 000000         NINPT1  003302R 025
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020          M.LIT  = 000205         NINPT2  003300R 025
F.BR   = 000020          F.STBK= 000036          LITOP   005072R 025 M.LN   = 000001         NINSTK  003304R 025
F.CHR  = 000075          F.STK  = 000022          LNLNG  = 104541          M.MMD  = 000010         NINTMP  003310R 025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R 025 M.NEG  = 000000         NINTYP  003306R 025
F.CODE= 000010          F.STP  = 000001          LOWCOR  000002R 025 M.NOT  = 000000         NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R 025 M.NUB  = 000006         NOTOPN= 104473
F.DSPT= 000044          F.TRC  = 000002          LWIDTH  000070R 025 M.NUM  = 000204         NPRT1   004762R 025
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003          M.OUTR= 000012         NULL    007364R 025
F.EFBK= 000010          F.URBD= 000020          L.CNT  = 000002          M.PITM= 000000         NUMCHN= 000020
F.EFN  = 000050          F.VAL  = 177776          L.PRNT= 000001          M.PRDS= 000006         NUMDFL  003230R 025
F.END  = 002000          F.VBN  = 000064          L.PTR  = 000000          M.QQAD= 000000         NUMEFL  003233R 025
F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004          M.QUAD= 000000         NUMESN  003236R 025
F.ERR  = 000052          GNOME   003102R 025 L.SUBS= 000002          M.RAND= 000012         NUMFLG  003230R 025
F.FACC= 000043          HASHC   005136R 025 MACHIN= 000106          M.RAVL= 000000         NUMLEN  000062R 025
F.FFBY= 000014          HASHTB  005140R 025 MAXCHN= 000014          M.RCIP= 000000         NUMNEG  003232R 025
F.FLAG= 000002          HSHMSK= 177601          MONBLK  003120R 025 M.RDCT= 000011         NUMQFL  003234R 025
F.FLB  = 000100          HSHSIZ= 000200          MONCVT  003132R 025 M.REVS= 000003         NUMRIG  003231R 025
F.FNAM= 000110          ICHAN   005050R 025 MONLEN  003130R 025 M.ROLL= 000000         NUMWID  003226R 025
```


SYMBOL TABLE

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 48-3
SYMBOL TABLE

NUMZFL  003235R     025 RANERR= 104411          R.NEXT= 000006          SY.BS = 000010          T.LIT = 000020
NXTIND= ****** GX       RARGBL  003204R  025 R.OFF = 000000          SY.CLR= 000040          T.MON = 000010
NYIERR= 104432          RARGCV  003220R  025 R.PREV= 000004          SY.ERP= 000020          T.NIL = 000004
N.BLSZ= 000036          RARGFL  003202R  025 R.PTR = 000000          SY.NOU= 000100          T.NUM = 002000
N.DESC= 000020          RARGLE  003216R  025 R.SEQ = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DEV = 000032          RARGLN  003210R  025 R.VAR = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEVN= 000034          RARGPT  003206R  025 R6    =%000006          SY.SRT= 000400          T.SOP = 000002
N.DID = 000024          RARGSF  003214R  025 R7    =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DVNM= 000032          RARGTY  003212R  025 SI    =%000005          SY.SVL= 000200          T.TMP = 000200
N.EXT = 000014          RARGVA  003222R  025 SIDAMN= 104433          S.AB  = 177776          T.VAL = 100000
N.FID = 000000          RBASE   005074R  025 SIFRAM  003314R  025 S.ASB = 100000          T.VAR = 001000
N.FNAM= 000006          RCOUNR  005112R  025 SISTAK  006350R  025 S.BFHD= 000020          UPPER = 177400
N.FTYP= 000014          RCOUNT  005110R  025 SITOP   006346R  025 S.BITS= 000002          UTNAME  000344R     023
N.FVER= 000016          REDCNT  005070R  025 SITP    007144R  025 S.CHR1= 000003          VALERR= 104413
N.NAME= 000006          REDIGN  005066R  025 SIWARN= 000200          S.FATT= 000016          VARBLE  007400R     025
N.NEXT= 000022          REDOFF  005072R  025 SIZTAB  000044R  025 S.FDB = 000140          VARBLS  000000R     025
N.PPN = 000024          REENTR  003060R  025 SI.BIT= 177774          S.FLB = 040000          VERNUM  000000R     025
N.STAT= 000020          REGTYP= 000001          SI.LEN= 177772          S.FNAM= 000006          VRSION= 000002
N.UNIT= 000034          RESBIT  003142R  025 SI.RET= 177772          S.FNB = 000036          WIDTH   000072R     025
OBUFLN= 001000          RESCVT  003152R  025 SI.VAL= 177776          S.FNBW= 000017          WROUND  000020R     025
OCHAN   005052R     025 RESERR= 104440          SKPSIZ  005134R  025 S.FNTY= 000004          WSDATE  000136R     025
OMODE   000402R     023 RESLEN  003150R  025 SPHOLD  000004R  024 S.FTYP= 000002          WSFULL= 104403
OPTMP0  005066R     025 RESPTR  003140R  025 SPINIT  000002R  024 S.LEN = 000002          WSNAME  000074R     025
OPTMP1  005070R     025 RESSFT  003146R  025 STKINC= 000774          S.LINK= 177774          WSTIME  000132R     025
OPTMP2  005072R     025 RESTYP  003144R  025 STOP  = 104420          S.NFEN= 000020          $FIS  = ****** G
OPTMP3  005074R     025 RESULT  003136R  025 SUBCNT  003104R  025 S.TYPE= 140000          $FPP  = ****** G
OPTMP4  005076R     025 RESVAL  003154R  025 SUBDCT  003106R  025 S.VAL = 000000          $FPPLN= ****** G
OPTMP5  005100R     025 RINC    005104R  025 SUBPTR  003110R  025 TKDIMS  005072R  025 $MACHN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R  025 SYMB1   005344R  025 TKFRAM  005066R  025 $RSTS = ****** G
OPTMP7  005104R     025 ROTINC  005070R  025 SYMF    005340R  025 TKLEFT  005070R  025 $RSX  = ****** G
OPTMP8  005106R     025 ROUND   000010R  025 SYMPT1  005346R  025 TMODE   000403R  023 $RT11 = ****** G
OPTMP9  005110R     025 ROVER   003312R  025 SYMTAB  005352R  025 TMPTAG= 000166          .BOOLE= 000002
OPTM10  005112R     025 RSTS  = 000000          SYMTBF= 104404      TTYBUF  004140R  025 .CHARA= 000000
OPTM14  005026R     025 RSX   = 000115          SYMTOP  005350R  025 TTYLEN= 000622          .DEF. = 000002
ORIGIN  000030R     025 RTOP    005070R  025 SYMTP   006272R  025 TURKEY= 104436          .FLOAT= 000006
OUTPTR  000066R     025 RT11  = 000000          SYMWRN= 000054      TYPE  = 000166          .FSRCB= ****** G
OVROPR= 000000          R.BITS= 000003          SYNERR= 104407      T.BOL = 020000          .INTEG= 000004
PATCH   007344R     025 R.FIX = 000001          SYSERR= 104402      T.CON = 004000          .MOLUN  000404R     023
PRECSN  000052R     025 R.LCT = 000002          SYSTAT  000000R  024 T.DIM = 000100          ...GBL= 000000
RANDAC  003064R     025 R.LN  = 000002          SY.BAD= 000001      T.FUN = 010000          ...TPC= 000020
RANDY   003062R     025 R.LNUM= 000010          SY.BRK= 100000      T.LIST= 000400

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```



```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 48-4
SYMBOL TABLE
```

J 15

```
ARYSET  MACRO M1113  08-FEB-80 10:44  PAGE 48-4
SYMBOL TABLE

MNTRTL  000000     017
CODE    000316     020
PATCH   000000     021
$$FSR1  000000     022
DEF$0   000406     023
DEF$1   000006     024
DEF$2   007400     025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12938 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:01
ARYSET,DM1:[305,301]ARYSET/-SP=APLRX1,APLDEF,ARYSET,END
```

K 15

```
**FILE**ID**ASSFIL
```

```
**FILE**ID**ASSFIL
```

```
  AAAAAA       SSSSSSSS     SSSSSSSS   FFFFFFFFFF     IIIIII     LL
  AAAAAA       SSSSSSSS     SSSSSSSS   FFFFFFFFFF     IIIIII     LL
AA      AA   SS           SS           FF               II       LL
AA      AA   SS           SS           FF               II       LL
AA      AA   SS           SS           FF               II       LL
AA      AA   SS           SS           FF               II       LL
AA      AA     SSSSSS       SSSSSS     FFFFFFFF         II       LL
AA      AA     SSSSSS       SSSSSS     FFFFFFFF         II       LL
AAAAAAAAAA           SS           SS   FF               II       LL
AAAAAAAAAA           SS           SS   FF               II       LL
AA      AA           SS           SS   FF               II       LL                .....
AA      AA           SS           SS   FF               II       LL                .....
AA      AA   SSSSSSSS     SSSSSSSS     FF             IIIIII     LLLLLLLLLL        .....
AA      AA   SSSSSSSS     SSSSSSSS     FF             IIIIII     LLLLLLLLLL        .....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```



```
.MAIN.  MACRO M1113  08-FEB-80 10:46
TABLE OF CONTENTS
```

.MAIN. MACRO M1113 08-FEB-80 10:46
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    COMMANDS                ASSIGN
```


ASSFIL MACRO M1113 08-FEB-80 10:46 PAGE 1

ASSFIL MACRO M1113 08-FEB-80 10:46 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```


ASSFIL MACRO M1113 08-FEB-80 10:46 PAGE 2
APLDEF

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

B 16

```
APL     MACRO M1113  08-FEB-80 10:46  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  S_Q,SRC,LOC,BIN,BEX,MEB
```

C 16

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 44
DEFINITIONS     ERRORS
```

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  ASSFIL
                                        .IDENT  \V02.03\
                                        .SBTTL  COMMANDS                ASSIGN
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       ASSIGN A FILE TO A PARTICULAR CHANNEL (DO NOT CREATE IF NOT THERE!!)
                                        ;--
000000                                          .PSECT  CODE

000000                                  ASSFIL::
000000                                          CALL    CMDCHN          ;GET CHANNEL NUMBER
000004  001401                                  BEQ     1$              ;MUST BE SPACE DELIMETER
000006  104464                                  ILFNAM                  ;ILLEGAL FILE NAME

000010  032762  010000  177756  1$:             BIT     #C.OPN,C.SBIT(R2) ;CHANNEL OPEN?
000016  001015                                  BNE     2$              ;CHANNEL IN USE
000020                                          CALL    RESCHN          ;Reset, just in case
000024  010200                                  MOV     R2,R0           ;Copy theFDB pointer
000026  010201                                  MOV     R2,R1           ;Create a pointer
000030  062701  000102                          ADD     #C.NAME,R1      ;    to the name block
000034  012702  000000G                         MOV     #DFLTEX,R2      ;Default File Name Block
000040                                          CALL    DECNAM          ;Get the file name
000044  005001                                  CLR     R1              ;CLEAR 'CREATE' FLAG
000046                                          CALLR   OPNSET          ;SET UP OPEN

000052  104471                          2$:     CHNOPN                  ;CHANNEL ALREADY OPEN
```

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46
COMMANDS                ASSIGN

      1          000001                              .END
```



```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46-1
SYMBOL TABLE
```

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.ALOC= 000162  D.CONT= 000006  EDICNT  005064R  025
AC1    =%000001          CHRLBR= 000073  C.BLOK= 000064  D.CTLM= 000000  EDIFLG  005062R  025
AC2    =%000002          CHRLPR= 000072  C.BLSZ= 000140  D.DEAL= 000003  EDIT   = 000003
AC3    =%000003          CHRMNS= 000137  C.BUFF= 177760  D.DECD= 000002  EIS    = 000000
AC4    =%000004          CHRMUL= 000075  C.BUFL= 177762  D.DIV = 000000  ELESIZ  005132R  025
AC5    =%000005          CHRNEG= 000100  C.CCNT= 177766  D.DMD = 000010  ENDCOR  000004R  025
ALLINT  003274R  025 CHRPDL= 000271  C.CPTR= 177764  D.DQAD= 177777  EOF    = 104475
ARYRET  003276R  025 CHRPLU= 000055  C.CZ   = 001000  D.DROP= 000002  ERRMAX= 000166
ASCQUO= 000042          CHRPWR= 000120  C.DIG  = 000040  D.ENCD= 000002  ERRNUM  004772R  025
ASSFIL  000000RG 020 CHRQST= 000121  C.FASW= 020000  D.EQ   = 000000  ERRPTR  004770R  025
A.ALEN= 000004          CHRQUO= 000113  C.FLEN= 000010  D.EXPD= 000002  ERRO    004774R  025
A.ARG = 000040          CHRRBR= 000047  C.FLSH= 004000  D.FGET= 000005  EXEBUF  005054R  025
A.ASB = 000200          CHRRPR= 000042  C.FRC  = 000001  D.FMT  = 000004  ESCT    005056R  025
A.BITS= 000003          CHRSEM= 000074  C.IDEN= 000010  D.FPNT= 000005  ESPT    005060R  025
A.CON = 000020          CHRSMO= 000112  C.NAME= 000102  D.FPUT= 000005  FA.APD= 000100
A.DIMS= 000002          CHRSPA= 000040  C.NOBF= 000400  D.GE   = 000000  FA.CRE= 000010
A.DIM1= 000006          CHRUPA= 000131  C.NUM  = 000020  D.GT   = 000000  FA.DLK= 001000
A.NUM = 000006          CHR.BS= 000010  C.OP   = 000004  D.IBM  = 000002  FA.ENB= 100000
A.PTR = 000000          CHR.CC= 000003  C.OPN  = 010000  D.IND  = 000002  FA.EXC= 002000
A.TMP = 000010          CHR.CR= 000015  C.RDLN= 177776  D.INNR= 000013  FA.EXT= 000004
BADDIR= 104463          CHR.CZ= 000032  C.REWT= 002000  D.INTR= 000006  FA.NSP= 000100
BADERR= 104437          CHR.GT= 000046  C.RND  = 000010  D.LE   = 000000  FA.NSR= 000200
BLZBUB  005116R  025 CHR.LF= 000012  C.SBIT= 177756  D.LOG  = 000001  FA.POS= 010000
BRSELF  005046R  025 CHR.LT= 000043  C.SPFX= 000100  D.LT   = 000000  FA.RD  = 000001
CCNT    005072R  025 CHR.SI= 000017  C.SPGL= 000200  D.MAX  = 000000  FA.RWD= 004000
CHANEX  003060R  025 CHR.SL= 000057  C.SPLX= 000001  D.MEM  = 000002  FA.SEQ= 040000
CHAN00  000256R  025 CHR.SO= 000016  C.SPRS= 000002  D.MIN  = 000000  FA.SHR= 000040
CHAN01  000330R  025 CHR.TB= 000011  C.SRC  = 177774  D.MUL  = 000000  FA.TMP= 000020
CHAN02  000512R  025 CHR.UL= 000106  C.SRCP= 000024  D.NE   = 000000  FA.WCK= 020000
CHAN03  000674R  025 CHR..A= 000141  C.TT   = 000004  D.NN   = 000000  FA.WRT= 000002
CHAN04  001056R  025 CHR..E= 000145  C.WIDC= 177770  D.NR   = 000000  FD.BLK= 000010
CHAN05  001240R  025 CHR..S= 000163  C.WIDR= 177772  D.OR   = 000000  FD.CCL= 000002
CHAN06  001422R  025 CHR..T= 000164  DEBUG  = 000000  D.PLUS= 000000  FD.COM= 020000
CHAN07  001604R  025 CHR..U= 000165  DECNAM= ****** GX  D.PSUB= 000006  FD.CR  = 000002
CHAN08  001766R  025 CHR..Z= 000172  DEFERR= 104405  D.PWR  = 000001  FD.DIR= 000010
CHAN09  002150R  025 CHR..0= 000060  DEVIL   005114R  025 D.QQAD= 000000  FD.FTN= 000001
CHAN10  002332R  025 CHR..1= 000061  DEVNTF= 104470  D.QUAD= 000000  FD.F11= 040000
CHAN11  002514R  025 CH.AND= 000001  DFLTEX= ****** GX  D.RES  = 000000  FD.INS= 000010
CHAN12  002676R  025 CMDBUF  003250R  025 DIGITS  000006R  025 D.RHO  = 000000  FD.ISP= 002000
CHAN13  000022R  023 CMDCHN= ****** GX  DIMAX  = 000020  D.ROT  = 000003  FD.MNT= 100000
CHAN14  000204R  023 CMDMAX= 000010  DIMBLK  003112R  025 D.SCON= 000006  FD.OSP= 004000
CHNOPN= 104471          CNTSIZ  005130R  025 DIMPTR  003114R  025 D.SDIF= 000006  FD.PLC= 000004
CHRAND= 000051          COMBIN  005076R  025 DIMTYP  003116R  025 D.SUB  = 000000  FD.PRN= 000004
CHRBEL= 000007          COMCNT  005126R  025 DIRFUL= 104466  D.SUBS= 000006  FD.PSE= 010000
CHRBKR= 000133          COMERR= 104425  DOMERR= 104417  D.TAKE= 000002  FD.RAH= 000001
CHRBOL= 000200          COMPTR  005122R  025 DPTERR= 104416  D.TRAN= 000004  FD.RAN= 000002
CHRBOX= 000114          COMRTN  003156R  025 DYDCNT  003224R  025 D.UNON= 000006  FD.REC= 000001
CHRCMA= 000054          COMSKP  005124R  025 D.AND  = 000000  D.$MAX= 000000  FD.RWM= 000001
CHRCOL= 000076          CORPTR  005120R  025 D.BXCT= 000007  D.$MIN= 000000  FD.SDI= 000020
CHRDEL= 000107          CS.ASS= 000200  D.BXIO= 000007  D.$MUL= 000000  FD.SQD= 000040
CHRDIV= 000053          CS.BOO= 000010  D.BXNL= 000007  D.$PLU= 000000  FD.TTY= 000004
CHRDLR= 000176          CS.FLO= 000020  D.BXPP= 000007  D.$RES= 000000  FD.WBH= 000002
CHRDLT= 000110          CS.IDN= 000040  D.BXPW= 000007  D.$SUB= 000000  FF.CHR= 000005
CHRDOT= 000056          CS.ONE= 000100  D.BXRL= 000007  D.1COM= 000001  FF.NV  = 000003
CHREP  = 000105         CTEPS   000152R  025 D.CIRC= 000001  D.1EXP= 000002  FF.POE= 000002
CHRERR= 104431          CTMAX   000212R  025 D.COMB= 000001  D.1ROT= 000003  FF.RWD= 000001
CHRFIL= 000177          CTRMAX  000172R  025 D.COMP= 000001  EDICHR  005063R  025 FF.RWF= 000006
```



```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46-2
SYMBOL TABLE
```

ASSFIL MACRO M1113 08-FEB-80 10:46 PAGE 46-2
SYMBOL TABLE

```
FF.SPC= 000004          F.FLAG= 000002  HSHMSK= 177601          MONCVT  003132R 025 M.REVS= 000003
FHEADR  005012R 025     F.FLB = 000100  HSHSIZ= 000200          MONLEN  003130R 025 M.ROLL= 000000
FHRET   005040R 025     F.FNAM= 000110  ICHAN   005050R 025     MONPTR  003122R 025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102  ILFNAM= 104464          MONSFT  003126R 025 M.SCAN= 000011
FIS    = 00000C         F.FNDF= 000010  ILLCHN= 104540          MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116  ILLSW = 104565          MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025     F.FUN = 000010  INDERR= 104410          M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025     F.FVER= 000120  INNTMP  003250R 025     M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025     F.HEAD= 000006  INNTM2  005016R 025     M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025     F.HIBK= 000004  INPTR   000064R 025     M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025     F.LBL = 000020  INSNAM= 104462          M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025     F.LINE= 000012  IOERR = 104477          M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025     F.LOC = 000004  I.CNT = 177776          M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004  I.DIM = 177772          M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025     F.LUN = 000042  I.LEN = 000012          M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025     F.MBCT= 000054  I.OERR= 000062          M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055  I.OLD = 177770          M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056  I.PTR = 177774          M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002  I.STR = 177766          M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001  LABERR= 104406          M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024  LABLPT  005042R 025     M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025     F.NREC= 000030  LARGBL  003162R 025     M.BXWA= 000007      NB.DIR= 000100
FPP    = 000000         F.OPN = 000010  LARGCV  003176R 025     M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030  LARGFL  003160R 025     M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025     F.PREV= 000000  LARGLE  003174R 025     M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025     F.PRNT= 100000  LARGLN  003166R 025     M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025     F.PROT= 010000  LARGPT  003164R 025     M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016  LARGSF  003172R 025     M.ENCD= 000004      NB.SVR= 000010
F$EVI  = 000200         F.RATT= 000001  LARGTY  003170R 025     M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034  LARGVA  003200R 025     M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017  LBINC   005100R 025     M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000  LBUFF   003316R 025     M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040  LBUFLN= 000621          M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040  LCOUNT  005106R 025     M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772  LENERR= 104412          M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002  LENTAB  000054R 025     M.GDWN= 000003      ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000  LEXPTR  003100R 025     M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014  LFTOP   005066R 025     M.IBM = 000003      ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016  LINC    005102R 025     M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020  LINE    005016R 025     M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100  LININC  005026R 025     M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072  LISIZE= 000020          M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074  LITOP   005072R 025     M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036  LNLNG = 104541          M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022  LOCLCT  005036R 025     M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004  LOWCOR  000002R 025     M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001  LSTERP  004766R 025     M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002  LWIDTH  000070R 025     M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002  L.BITS= 000003          M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136  L.CNT = 000002          M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020  L.PRNT= 000001          M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776  L.PTR = 000000          M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064  L.PTR1= 000004          M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060  L.SUBS= 000002          M.RAND= 000012      NUMFLG  003230R 025
F.ERR = 000052          GNOME   003102R 025 MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025 MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011      NUMQFL  003234R 025
```


ASSFIL MACRO M1113 08-FEB-80 10:46 PAGE 46-3
SYMBOL TABLE

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46-3
SYMBOL TABLE

NUMRIG  003231R     025 RANDY   003062R     025 R.LNUM= 000010          SY.BRK= 100000          T.LIST= 000400
NUMWID  003226R     025 RANERR= 104411          R.NEXT= 000006          SY.BS = 000010          T.LIT = 000020
NUMZFL  003235R     025 RARGBL  003204R     025 R.OFF = 000000          SY.CLR= 000040          T.MON = 000010
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL = 000004
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR = 000000          SY.NOU= 000100          T.NUM = 002000
N.DESC= 000020          RARGLE  003216R     025 R.SEQ = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DEV = 000032          RARGLN  003210R     025 R.VAR = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEVN= 000034          RARGPT  003206R     025 R6    =%000006          SY.SRT= 000400          T.SOP = 000002
N.DID = 000024          RARGSF  003214R     025 R7    =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DVNM= 000032          RARGTY  003212R     025 SI    =%000005          SY.SVL= 000200          T.TMP = 000200
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433          S.AB  = 177776          T.VAL = 100000
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R  025 S.ASB = 100000          T.VAR = 001000
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R  025 S.BFHD= 000020          UPPER = 177400
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R  025 S.BITS= 000002          UTNAME  000344R     023
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R  025 S.CHR1= 000003          VALERR= 104413
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200          S.FATT= 000016          VARBLF  007400R     025
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R  025 S.FDB = 000140          VARBLS  000000R     025
N.PPN = 000024          REENTR  003060R     025 SI.BIT= 177774          S.FLB = 040000          VERNUM  000000R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772          S.FNAM= 000006          VRSION= 000002
N.UNIT= 000034          RESCHN= ****** GX       SI.RET= 177772          S.FNB = 000036          WIDTH   000072R     025
OBUFLN= 001000          RESCVT  003152R     025 SI.VAL= 177776          S.FNBW= 000017          WROUND  000020R     025
OCHAN   005052R     025 RESERR= 104440          SKPSIZ  005134R  025 S.FNTY= 000004          WSDATE  000136R     025
OMODE   000402R     023 RESLEN  003150R     025 SPHOLD  000004R  024 S.FTYP= 000002          WSFULL= 104403
OPNSET= ****** GX       RESPTR  003140R     025 SPINIT  000002R  024 S.LEN = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESSFT  003146R     025 STKINC= 000774          S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESTYP  003144R     025 STOP  = 104420          S.NFEN= 000020          $FIS  = ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBCNT  003104R  025 S.TYPE= 140000          $FPP  = ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBDCT  003106R  025 S.VAL = 000000          $FPPLN= ****** G
OPTMP4  005076R     025 RINC    005104R     025 SUBPTR  003110R  025 TKDIMS  005072R  025 $MACHN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMB1   005344R  025 TKFRAM  005066R  025 $RSTS = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMF    005340R  025 TKLEFT  005070R  025 $RSX  = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMPT1  005346R  025 TMODE   000403R  023 $RT11 = ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMTAB  005352R  025 TMPTAG= 000166          .BOOLE= 000002
OPTMP9  005110R     025 RSTS  = 000000          SYMTBF= 104404          TTYBUF  004140R  025 .CHARA= 000000
OPTM10  005112R     025 RSX   = 000115          SYMTOP  005350R  025 TTYLEN= 000622          .DEF. = 000002
OPTM14  005026R     025 RTOP    005070R     025 SYMTP   006272R  025 TURKEY= 104436          .FLOAT= 000006
ORIGIN  000030R     025 RT11  = 000000          SYMWRN= 000054          TYPE  = 000166          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003          SYNERR= 104407          T.BOL = 020000          .INTEG= 000004
OVROPR= 000000          R.FIX = 000001          SYSERR= 104402          T.CON = 004000          .MOLUN  000404R     023
PATCH   007344R     025 R.LCT = 000002          SYSTAT  000000R  024 T.DIM = 000100          ...GBL= 000000
PRECSN  000052R     025 R.LN  = 000002          SY.BAD= 000001          T.FUN = 010000          ...TPC= 000020
RANDAC  003064R     025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
```

H 16

```
ASSFIL  MACRO M1113  08-FEB-80 10:46  PAGE 46-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000054      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12924 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:49
ASSFIL,DM1:[305,301]ASSFIL/-SP=APLRX1,APLDEF,ASSFIL,END
```

I 16

```
**FILE**ID**ASSIGN
```

```
**FILE**ID**ASSIGN
```

```
  AAAAAA       SSSSSSSS     SSSSSSSS     IIIIII       GGGGGGGG   NN      NN
  AAAAAA       SSSSSSSS     SSSSSSSS     IIIIII       GGGGGGGG   NN      NN
AA      AA   SS           SS             II         GG           NN      NN
AA      AA   SS           SS             II         GG           NN      NN
AA      AA   SS           SS             II         GG           NNNN    NN
AA      AA   SS           SS             II         GG           NNNN    NN
AA      AA     SSSSSS       SSSSSS       II         GG           NN  NN  NN
AA      AA     SSSSSS       SSSSSS       II         GG           NN  NN  NN
AAAAAAAAAA           SS           SS     II         GG  GGGGGG   NN    NNNN
AAAAAAAAAA           SS           SS     II         GG  GGGGGG   NN    NNNN
AA      AA           SS           SS     II         GG      GG   NN      NN      ....
AA      AA           SS           SS     II         GG      GG   NN      NN      ....
AA      AA   SSSSSSSS     SSSSSSSS       IIIIII       GGGGGG     NN      NN      ....
AA      AA   SSSSSSSS     SSSSSSSS       IIIIII       GGGGGG     NN      NN      ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```



```
.MAIN.  MACRO M1113  08-FEB-80 10:48
TABLE OF CONTENTS
```

.MAIN. MACRO M1113 08-FEB-80 10:48
TABLE OF CONTENTS


```
     2-    1     APLDEF
     2-    8     A P L
     2-    9
     2-   10     A PROGRAMMING LANGUAGE
     2-   11
     2-   12     STEVE POULSEN & DAVE VANN
     2-   13     11-Jul-76
     2-   14     ROBERT GOTTLIEB & PETER HAYNES
     2-   15     2-Feb-79
     2-   16
     2-   17     V2
     2-   18
     2-   19
     2-   20     COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25     COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26     COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27     BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28     USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29     TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30     DEC.
     2-   31
     2-   32     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33     AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34     CORPORATION.
     2-   35
     2-   36     DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37     SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46     DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139     DEFINITIONS      OPERATOR OVERLAYS
     5-  268     DEFINITIONS      REGISTERS
     6-  331     DEFINITIONS      MEMORY ORGANIZATION
     7-  377     DEFINITIONS      ORDERING PSECTS
     8-  426     DEFINITIONS      SI PARSE FRAMES
     9-  474     DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515     DEFINITIONS      SI EXECUTION FRAMES
    11-  580     DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607     DEFINITIONS      CHRTAB BITS
    12-  624     DEFINITIONS      SYSTEM STATUS BITS
    13-  640     DEFINITIONS      F-FRAMES
    14-  724     DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807     DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869     DEFINITIONS      LIST STORAGE BLOCK
    18-  911     DEFINITIONS      INDEXING STACK FRAME
    19-  940     DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969     DEFINITIONS      CHANNEL BLOCK
    22- 1085     DEFINITIONS              RSX DATA
    23- 1110     DEFINITIONS      NAME BLOCKS
    33- 1612     DEFINITIONS      MACROS
    35- 1905     DEFINITIONS      APL CHARACTERS
    36- 1967     DEFINITIONS      ERRORS
    44-    4     ASSIGN           BIND VALUES TO VARIABLES
```


ASSIGN MACRO M1113 08-FEB-80 10:48 PAGE 1

```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 1


      1  000000                RT11    =  0          ;NOT RT-11
      2  000000                RSTS    =  0          ;NOT RSTS/E
      3  000115                RSX     =  'M         ;RSX-11/M
      4
      5  000000                EIS     =  0          ;NO EIS
      6  000000                FIS     =  0          ;NO FIS
      7  000000                FPP     =  0          ;NO FPP
      8
      9  000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11  000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```



```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 2
APLDEF
```

```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```



```
APL     MACRO M1113  08-FEB-80 10:48  PAGE 42
DEFINITIONS     ERRORS
```

```
APL     MACRO M1113  08-FEB-80 10:48  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

B	1			J	5			E	10			M	14		
C	1			K	5			F	10			N	14		
D	1			L	5			G	10			B	15		
E	1			M	5			H	10			C	15		
F	1			N	5			I	10			D	15		
G	1			B	6			J	10			E	15		
H	1			C	6			K	10			F	15		
I	1			D	6			L	10			G	15		
J	1			E	6			M	10			H	15		
K	1			F	6			N	10			I	15		
L	1			G	6			B	11			J	15		
M	1			H	6			C	11			K	15	ASSFIL	
N	1			I	6			D	11			L	15		
B	2			J	6			E	11			M	15		
C	2			K	6			F	11			N	15		
D	2			L	6			G	11			B	16		
E	2			M	6			H	11			C	16		
F	2			N	6			I	11			D	16		
G	2			B	7			J	11			E	16		
H	2			C	7			K	11			F	16		
I	2			D	7			L	11			G	16		
J	2			E	7			M	11			H	16		
K	2			F	7			N	11			I	16	ASSIGN	
L	2			G	7			B	12			J	16		
M	2			H	7			C	12			K	16		
N	2			I	7			D	12			L	16		
B	3			J	7			E	12			M	16		
C	3			K	7			F	12						
D	3			L	7			G	12						
E	3			M	7			H	12						
F	3	APL010		N	7			I	12						
G	3			B	8			J	12						
H	3			C	8			K	12						
I	3			D	8			L	12						
J	3			E	8			M	12						
K	3			F	8	APL011		N	12						
L	3			G	8			B	13						
M	3			H	8			C	13						
N	3			I	8			D	13						
B	4			J	8			E	13						
C	4			K	8			F	13						
D	4			L	8			G	13						
E	4			M	8			H	13	APRINT					
F	4			N	8			I	13						
G	4			B	9			J	13						
H	4			C	9			K	13						
I	4			D	9			L	13						
J	4			E	9			M	13						
K	4			F	9			N	13						
L	4			G	9			B	14						
M	4			H	9			C	14						
N	4			I	9			D	14						
B	5			J	9			E	14						
C	5			K	9			F	14						
D	5			L	9			G	14						
E	5			M	9			H	14						
F	5			N	9			I	14						
G	5			B	10			J	14						
H	5			C	10			K	14	ARYSET					
I	5			D	10			L	14						

```
                                .TITLE  ASSIGN
                                .IDENT  \V02.03\

                                .SBTTL  ASSIGN          BIND VALUES TO VARIABLES

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'ASSIGN' assigns values to variables.  A syntax
                                ;       error results if the value is not a val or if the
                                ;       symbol is not a variable (a function perhaps?).  If
                                ;       the variable had a previous value, it is deallocated.
                                ;       if the val is not temporary or is a codestring constant,
                                ;       the array is copied so that if a subscripted assignment
                                ;       (which is what this routine also does) is made
                                ;       nothing else gets screwed up.
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     #<SI frame>,R1
                                ;       MOV     #<Variable name>,R4
                                ;       CALL    ASSIGN
                                ;
                                ; INPUTS:
                                ;       R1      - Points to SI frame containing value.
                                ;       R4      - Points to variable name
                                ;
                                ; IMPLICIT INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       RETCOR  - Used to delete old value
                                ;       GETSAM  - Used to get room for new array
                                ;       ACOPY   - Used to copy the new value to the variable
                                ;
                                ; SIDE EFFECTS:
                                ;       Assigns the value in the frame pointed to by R1
                                ;       to the variable pointed to by R4.
                                ;--

000000                          .PSECT CODE

000000                          ASSIGN::
000000  042737  10C000  004764' 1$:     BIC     #F.PRNT,FLAGS   ;DON'T PRINT RESULT
000006  032761  100000  177774          BIT     #T.VAL,SI.BIT(R1);VAL ON SI?
000014  001001                          BNE     2$              ;YES, GOOD
000016  104407                          SYNERR                  ;NO, BAD
000020                          2$:     PUSH    R2              ;SAVE INPUT CHAR POINTER
000022  011400                          MOV     @R4,R0          ;GET OLD VALUE
000024  001422                          BEQ     3$              ;IF IT HAD ANY
000026  005760  000002                  TST     A.DIMS(R0)      ;CHECK FOR TYPE
000032  002371                          BGE     1$              ;YES, CAN'T KILL FUNCTION OR GROUP!
000034  020061  177776                  CMP     R0,SI.VAL(R1)   ;ASSIGNING SELF TO SELF??
```

```
     58 000040  001424                          BEQ     4$                  ;YES, HANDLE SPECIAL CASE!!!!!!(GASP!)
     59 000042  116002  000003                  MOVB    A.BITS(R0),R2       ;Get BITS Field
     60 000046  132702  000020                  BITB    #A.CON,R2           ;A Label?
     61 000052  001043                          BNE     6$                  ;Yes, is an Error
     62 000054  142702  000010                  BICB    #A.TMP,R2           ;VALUE IS NOW TEMPORARY
     63 000060  132702  000040                  BITB    #A.ARG,R2           ;IS THIS A FUNCTION ARGUMENT?
     64 000064  001002                          BNE     3$                  ;YES, SO DON'T KILL IT!
     65 000066                                  CALL    RETCOR              ;RELEASE OLD VALUE
     66 000072  005014                  3$:     CLR     (R4)                ;NOTHING THERE
     67 000074  016100  177776                  MOV     SI.VAL(R1),R0       ;POINT TO NEW VALUE
     68 000100  132760  000070  000003          BITB    #A.CON+A.TMP+A.ARG,A.BITS(R0);NICE VALUE?
     69 000106  001007                          BNE     5$                  ;NO, COPY IT
     70 000110  010014                          MOV     R0,@R4              ;SET NEW VALUE FOR VAR
     71 000112                          4$:     POP     R2                  ;GET BACK INPUT POINTER
     72 000114  011400                          MOV     @R4,R0              ;GET REAL POINTER TO SYMBOL'S VALUE
     73 000116  152760  000010  000003          BISB    #A.TMP,A.BITS(R0);MARK AS NOT A TEMP
     74 000124                                  RETURN                      ;ALL DONE
     75
     76                                 ;VAL IS NOT A TEMP OR IS A CODE STRING CONSTANT.
     77                                 ;ALLOCATE A NEW BLOCK AND COPY IN THE ARRAY.
     78
     79 000126  016100  177776          5$:     MOV     SI.VAL(R1),R0       ;POINT TO VAL
     80 000132                                  PUSH    R0                  ;SAVE PTR TO ARRAY
     81 000134                                  CALL    GETSAM              ;GET ROOM FOR NEW ARRAY
     82 000140  062705  177772                  ADD     #SI.LEN,SI          ;POP POINTER TO NEW BLOCK
     83 000144  010014                          MOV     R0,@R4              ;SET NEW VALUE FOR VARIABLE
     84 000146                                  POP     R2                  ;POINT TO ARRAY TO COPY
     85 000150  016200  000004                  MOV     A.ALEN(R2),R0       ;GET LENGTH TO COPY
     86 000154                                  CALL    ACOPY               ;COPY ARRAY
     87 000160  000754                          BR      4$                  ;ALL DONE
     88
     89 000162  104406                  6$:     LABERR                      ;Label Error
```

```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 46
ASSIGN           BIND VALUES TO VARIABLES

      1          000001                          .END
```

```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 46-1
SYMBOL TABLE

ACOPY = ****** GX        CHRFIL= 000177        C.ALOC= 000162        D.DEAL= 000003        EDIT  = 000003
AC0   =%000000           CHRFRO= 000135        C.BLOK= 000064        D.DECD= 000002        EIS   = 000000
AC1   =%000001           CHRLBR= 000073        C.BLSZ= 000140        D.DIV = 000000        ELESIZ 005132R     025
AC2   =%000002           CHRLPR= 000072        C.BUFF= 177760        D.DMD = 000010        ENDCOR 000004R     025
AC3   =%000003           CHRMNS= 000137        C.BUFL= 177762        D.DQAD= 177777        EOF   = 104475
AC4   =%000004           CHRMUL= 000075        C.CCNT= 177766        D.DROP= 000002        ERRMAX= 000166
AC5   =%000005           CHRNEG= 000100        C.CPTR= 177764        D.ENCD= 000002        ERRNUM 004772R     025
ALLINT  003274R     025  CHRPDL= 000271        C.CZ  = 001000        D.EQ  = 000000        ERRPTR 004770R     025
ARYRET  003276R     025  CHRPLU= 000055        C.DIG = 000040        D.EXPD= 000002        ERRO   004774R     025
ASCQUO= 000042           CHRPWR= 000120        C.FASW= 020000        D.FGET= 000005        EXEBUF 005054R     025
ASSIGN  000000RG    020  CHRQST= 000121        C.FLEN= 000010        D.FMT = 000004        E$CT   005056R     025
A.ALEN= 000004           CHRQUO= 000113        C.FLSH= 004000        D.FPNT= 000005        E$PT   005060R     025
A.ARG = 000040           CHRRBR= 000047        C.FRC = 000001        D.FPUT= 000005        FA.APD= 000100
A.ASB = 000200           CHRRPR= 000042        C.IDEN= 000010        D.GE  = 000000        FA.CRE= 000010
A.BITS= 000003           CHRSEM= 000074        C.NAME= 000102        D.GT  = 000000        FA.DLK= 001000
A.CON = 000020           CHRSMO= 000112        C.NOBF= 000400        D.IBM = 000002        FA.ENB= 100000
A.DIMS= 000002           CHRSPA= 000040        C.NUM = 000020        D.IND = 000002        FA.EXC= 002000
A.DIM1= 000006           CHRUPA= 000131        C.OP  = 000004        D.INNR= 000013        FA.EXT= 000004
A.NUM = 000006           CHR.BS= 000010        C.OPN = 010000        D.INTR= 000006        FA.NSP= 000100
A.PTR = 000000           CHR.CC= 000003        C.RDLN= 177776        D.LE  = 000000        FA.NSR= 000200
A.TMP = 000010           CHR.CR= 000015        C.REWT= 002000        D.LOG = 000001        FA.POS= 010000
BADDIR= 104463           CHR.CZ= 000032        C.RND = 000010        D.LT  = 000000        FA.RD = 000001
BADERR= 104437           CHR.GT= 000046        C.SBIT= 177756        D.MAX = 000000        FA.RWD= 004000
BLZBUB  005116R     025  CHR.LF= 000012        C.SPFX= 000100        D.MEM = 000002        FA.SEQ= 040000
BRSELF  005046R     025  CHR.LT= 000043        C.SPGL= 000200        D.MIN = 000000        FA.SHR= 000040
CCNT    005072R     025  CHR.SI= 000017        C.SPLX= 000001        D.MUL = 000000        FA.TMP= 000020
CHANEX  003060R     025  CHR.SL= 000057        C.SPRS= 000002        D.NE  = 000000        FA.WCK= 020000
CHAN00  000256R     025  CHR.SO= 000016        C.SRC = 177774        D.NN  = 000000        FA.WRT= 000002
CHAN01  000330R     025  CHR.TB= 000011        C.SRCP= 000024        D.NR  = 000000        FD.BLK= 000010
CHAN02  000512R     025  CHR.UL= 000106        C.TT  = 000004        D.OR  = 000000        FD.CCL= 000002
CHAN03  000674R     025  CHR..A= 000141        C.WIDC= 177770        D.PLUS= 000000        FD.COM= 020000
CHAN04  001056R     025  CHR..E= 000145        C.WIDR= 177772        D.PSUB= 000006        FD.CR = 000002
CHAN05  001240R     025  CHR..S= 000163        DEBUG = 000000        D.PWR = 000001        FD.DIR= 000010
CHAN06  001422R     025  CHR..T= 000164        DEFERR= 104405        D.QQAD= 000000        FD.FTN= 000001
CHAN07  001604R     025  CHR..U= 000165        DEVIL   005114R  025  D.QUAD= 000000        FD.F11= 040000
CHAN08  001766R     025  CHR..Z= 000172        DEVNTF= 104470        D.RES = 000000        FD.INS= 000010
CHAN09  002150R     025  CHR..0= 000060        DIGITS  000006R  025  D.RHO = 000000        FD.ISP= 002000
CHAN10  002332R     025  CHR..1= 000061        DIMAX = 000020        D.ROT = 000003        FD.MNT= 100000
CHAN11  002514R     025  CH.AND= 000001        DIMBLK  003112R  025  D.SCON= 000006        FD.OSP= 004000
CHAN12  002676R     025  CMDBUF  003250R  025  DIMPTR  003114R  025  D.SDIF= 000006        FD.PLC= 000004
CHAN13  000022R     023  CMDMAX= 000010        DIMTYP  003116R  025  D.SUB = 000000        FD.PRN= 000004
CHAN14  000204R     023  CNTSIZ  005130R  025  DIRFUL= 104466        D.SUBS= 000006        FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R  025  DOMERR= 104417        D.TAKE= 000002        FD.RAH= 000001
CHRAND= 000051           COMCNT  005126R  025  DPTERR= 104416        D.TRAN= 000004        FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425        DYDCNT  003224R  025  D.UNON= 000006        FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R  025  D.AND = 000000        D.$MAX= 000000        FD.RWM= 000001
CHRBOL= 000200           COMRTN  003150R  025  D.BXCT= 000007        D.$MIN= 000000        FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R  025  D.BXIO= 000007        D.$MUL= 000000        FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R  025  D.BXNL= 000007        D.$PLU= 000000        FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200        D.BXPP= 000007        D.$RES= 000000        FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010        D.BXPW= 000007        D.$SUB= 000000        FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020        D.BXRL= 000007        D.1COM= 000001        FF.NV = 000003
CHRDLR= 000176           CS.IDN= 000040        D.CIRC= 000001        D.1EXP= 000002        FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100        D.COMB= 000001        D.1ROT= 000003        FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R  025  D.COMP= 000001        EDICHR  005063R  025  FF.RWF= 000006
CHREP = 000105           CTMAX   000212R  025  D.CONT= 000006        EDICNT  005064R  025  FF.SPC= 000004
CHRERR= 104431           CTRMAX  000172R  025  D.CTLM= 000000        EDIFLG  005062R  025  FHEADR  005012R     025
```

ASSIGN MACRO M1113 08-FEB-80 10:48 PAGE 46-2
SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110        HSHSIZ= 000200        MONLEN  003130R     025 M.ROLL= 000000
FILNTF= 104467          F.FNB = 000102        ICHAN   005050R 025   MONPTR  003122R     025 M.SATA= 000001
FIS   = 000000          F.FNDF= 000010        ILFNAM= 104464        MONSFT  003126R     025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116        ILLCHN= 104540        MONTYP  003124R     025 M.SHAP= 000000
FIXPTR  003074R     025 F.FUN = 000010        ILLSW = 104565        MONVAL  003134R     025 M.SIGN= 000000
FLAGS   004764R     025 F.FVER= 000120        INDERR= 104410        M.ABS = 000000          M.SUBR= 000001
FLCBUF  003250R     025 F.HEAD= 000006        INNTMP  003250R 025   M.BOL = 000200          M.SYM = 000203
FLINE   005014R     025 F.HIBK= 000004        INNTM2  005016R 025   M.BXAV= 000007          M.SYN = 000201
FLOEXP  003270R     025 F.LBL = 000020        INPTR   000064R 025   M.BXCR= 000007          M.TRAN= 000012
FLONEG  003272R     025 F.LINE= 000012        INSNAM= 104462        M.BXCT= 000007          M.$ABS= 000000
FLSWTB  000222R     025 F.LOC = 000004        IOERR = 104477        M.BXEX= 000007          M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 000004        I.CNT = 177776        M.BXFX= 000007          M.$FLR= 000000
FNINSI  005044R     025 F.LUN = 000042        I.DIM = 177772        M.BXIO= 000007          M.$IDN= 000000
FORIGN  000032R     025 F.MBCT= 000054        I.LEN = 000012        M.BXLC= 000003          M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055        I.OERR= 000062        M.BXNC= 000007          M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056        I.OLD = 177770        M.BXNL= 000007          M.1RDT= 000011
FO.RD = 000001          F.MON = 000002        I.PTR = 177774        M.BXPP= 000007          M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001        I.STR = 177766        M.BXPW= 000007          M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024        LABERR= 104406        M.BXRL= 000007          NAMBUF  003240R     025
FPMODE  000042R     025 F.NREC= 000030        LABLPT  005042R 025   M.BXTS= 000007          NB.DEV= 000200
FPP   = 000000          F.OPN = 000010        LARGBL  003162R 025   M.BXWA= 000007          NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030        LARGCV  003176R 025   M.CEI = 177777          NB.NAM= 000004
FREBLK  007400R     025 F.PREV= 000000        LARGFL  003160R 025   M.CHR = 000202          NB.SD1= 000400
FUZZ    000142R     025 F.PRNT= 100000        LARGLE  003174R 025   M.CMT = 000206          NB.SD2= 001000
FXPTR1  003076R     025 F.PROT= 010000        LARGLN  003166R 025   M.DOUT= 000006          NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016        LARGPT  003164R 025   M.DQAD= 000006          NB.STP= 000020
F$EVI = 000200          F.RATT= 000001        LARGSF  003172R 025   M.ENCD= 000004          NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034        LARGTY  003170R 025   M.EXE = 000000          NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017        LARGVA  003200R 025   M.EXP = 000001          NB.VER= 000001
F.ACTL= 000076          F.READ= 004000        LBINC   005100R 025   M.FACT= 000001          ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040        LBUFF   003316R 025   M.FIL = 000177          ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040        LBUFLN= 000621        M.FLR = 177777          ND.NAM= 000004
F.BDB = 000070          F.RET = 177772        LCOUNT  005106R 025   M.FMT = 000004          ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002        LENERR= 104412        M.GDUP= 000003          ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000        LENTAB  000054R 025   M.GDWN= 000003          ND.$E$= 000020
F.BKDN= 000026          F.R2  = 000014        LEXPTR  003100R 025   M.GETI= 000006          ND.$NM= 000000
F.BKDS= 000020          F.R3  = 000016        LFTOP   005066R 025   M.IBM = 000003          ND.$N$= 000040
F.BKEF= 000050          F.R4  = 000020        LINC    005102R 025   M.IDN = 000000          ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100        LINE    005016R 025   M.INSP= 000006          NEGSGN  005063R     025
F.BKST= 000024          F.SPDV= 000072        LININC  005026R 025   M.IOTA= 000000          NINPT1  003302R     025
F.BKVB= 000064          F.SPUN= 000074        LISIZE= 000020        M.LIT = 000205          NINPT2  003300R     025
F.BR  = 000020          F.STBK= 000036        LITOP   005072R 025   M.LN  = 000001          NINSTK  003304R     025
F.CHR = 000075          F.STK = 000022        LNLNG = 104541        M.MMD = 000010          NINTMP  003310R     025
F.CNTG= 000034          F.STOP= 000004        LOCLCT  005036R 025   M.NEG = 000000          NINTYP  003306R     025
F.CODE= 000010          F.STP = 000001        LOWCOR  000002R 025   M.NOT = 000000          NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002        LSTERP  004766R 025   M.NUB = 000006          NOTOPN= 104473
F.DSPT= 000044          F.TRC = 000002        LWIDTH  000070R 025   M.NUM = 000204          NPRT1   004762R     025
F.DVNM= 000134          F.UNIT= 000136        L.BITS= 000003        M.OUTR= 000012          NULL    007364R     025
F.EFBK= 000010          F.URBD= 000020        L.CNT = 000002        M.PITM= 000000          NUMCHN= 000020
F.EFN = 000050          F.VAL = 177776        L.PRNT= 000001        M.PRDS= 000006          NUMDFL  003230R     025
F.END = 002000          F.VBN = 000064        L.PTR = 000000        M.QQAD= 000000          NUMEFL  003233R     025
F.EOBB= 000032          F.VBSZ= 000060        L.PTR1= 000004        M.QUAD= 000000          NUMESN  003236R     025
F.ERR = 000052          GETSAM= ****** GX     L.SUBS= 000002        M.RAND= 000012          NUMFLG  003230R     025
F.FACC= 000043          GNOME   003102R 025   MACHIN= 000106        M.RAVL= 000000          NUMLEN  000062R     025
F.FFBY= 000014          HASHC   005136R 025   MAXCHN= 000014        M.RCIP= 000000          NUMNEG  003232R     025
F.FLAG= 000002          HASHTB  005140R 025   MONBLK  003120R 025   M.RDCT= 000011          NUMQFL  003234R     025
F.FLB = 000100          HSHMSK= 177601        MONCVT  003132R 025   M.REVS= 000003          NUMRIG  003231R     025
```

SYMBOL TABLE

```
NUMWID  003226R     025 RANERR= 104411          R.NEXT= 000006      SY.BS  = 000010      T.LIT  = 000020
NUMZFL  003235R     025 RARGBL  003204R    025 R.OFF  = 000000      SY.CLR= 000040      T.MON  = 000010
NYIERR= 104432          RARGCV  003220R    025 R.PREV= 000004      SY.ERP= 000020      T.NIL  = 000004
N.BLSZ= 000036          RARGFL  003202R    025 R.PTR  = 000000      SY.NOU= 000100      T.NUM  = 002000
N.DESC= 000020          RARGLE  003216R    025 R.SEQ  = 000003      SY.OPN= 000002      T.OPFN= 000001
N.DEV = 000032          RARGLN  003210R    025 R.VAR  = 000002      SY.PMT= 001000      T.OTHR= 040000
N.DEVN= 000034          RARGPT  003206R    025 R6     =%000006      SY.SRT= 000400      T.SOP  = 000002
N.DID = 000024          RARGSF  003214R    025 R7     =%000007      SY.SVC= 000004      T.SPOP= 000040
N.DVNM= 000032          RARGTY  003212R    025 SI     =%000005      SY.SVL= 000200      T.TMP  = 000200
N.EXT = 000014          RARGVA  003222R    025 SIDAMN= 104433       S.AB   = 177776      T.VAL  = 100000
N.FID = 000000          RBASE   005074R    025 SIFRAM  003314R  025 S.ASB  = 100000      T.VAR  = 001000
N.FNAM= 000006          RCOUNR  005112R    025 SISTAK  006350R  025 S.BFHD= 000020      UPPER  = 177400
N.FTYP= 000014          RCOUNT  005110R    025 SITOP   006346R  025 S.BITS= 000002      UTNAME  000344R     023
N.FVER= 000016          REDCNT  005070R    025 SITP    007144R  025 S.CHR1= 000003      VALERR= 104413
N.NAME= 000006          REDIGN  005066R    025 SIWARN= 000200       S.FATT= 000016      VARBLE  007400R     025
N.NEXT= 000022          REDOFF  005072R    025 SIZTAB  000044R  025 S.FDB  = 000140      VARBLS  000000R     025
N.PPN = 000024          REENTR  003060R    025 SI.BIT= 177774       S.FLB  = 040000      VERNUM  000000R     025
N.STAT= 000020          RESBIT  003142R    025 SI.LEN= 177772       S.FNAM= 000006      VRSION= 000002
N.UNIT= 000034          RESCVT  003152R    025 SI.RET= 177772       S.FNB  = 000036      WIDTH   000072R     025
OBUFLN= 001000          RESERR= 104440          SI.VAL= 177776       S.FNBW= 000017      WROUND  000020R     025
OCHAN   005052R     025 RESLEN  003150R    025 SKPSIZ  005134R  025 S.FNTY= 000004      WSDATE  000136R     025
OMODE   000402R     023 RESPTR  003140R    025 SPHOLD  000004R  024 S.FTYP= 000002      WSFULL= 104403
OPTMP0  005066R     025 RESSFT  003146R    025 SPINIT  000002R  024 S.LEN  = 000002      WSNAME  000074R     025
OPTMP1  005070R     025 RESTYP  003144R    025 STKINC= 000774       S.LINK= 177774      WSTIME  000132R     025
OPTMP2  005072R     025 RESULT  003136R    025 STOP   = 104420      S.NFEN= 000020      $FIS   = ****** G
OPTMP3  005074R     025 RESVAL  003154R    025 SUBCNT  003104R  025 S.TYPE= 140000      $FPP   = ****** G
OPTMP4  005076R     025 RETCOR= ****** GX       SUBDCT  003106R  025 S.VAL  = 000000      $FPPLN= ****** G
OPTMP5  005100R     025 RINC    005104R    025 SUBPTR  003110R  025 TKDIMS  005072R  025 $MACHN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R    025 SYMB1   005344R  025 TKFRAM  005066R  025 $RSTS  = ****** G
OPTMP7  005104R     025 ROTINC  005070R    025 SYMF    005340R  025 TKLEFT  005070R  025 $RSX   = ****** G
OPTMP8  005106R     025 ROUND   000010R    025 SYMPT1  005346R  025 TMODE   000403R  023 $RT11  = ****** G
OPTMP9  005110R     025 ROVER   003312R    025 SYMTAB  005352R  025 TMPTAG= 000166       .BOOLE= 000002
OPTM10  005112R     025 RSTS   = 000000        SYMTBF= 104404       TTYBUF  004140R  025 .CHARA= 000000
OPTM14  005026R     025 RSX    = 000115        SYMTOP  005350R  025 TTYLEN= 000622       .DEF.  = 000002
ORIGIN  000030R     025 RTOP    005070R    025 SYMTP   006272R  025 TURKEY= 104436       .FLOAT= 000006
OUTPTR  000066R     025 RT11   = 000000        SYMWRN= 000054       TYPE   = 000166      .FSRCB= ****** G
OVROPR= 000000          R.BITS= 000003          SYNERR= 104407       T.BOL  = 020000      .INTEG= 000004
PATCH   007344R     025 R.FIX  = 000001        SYSERR= 104402       T.CON  = 004000      .MOLUN  000404R     023
PRECSN  000052R     025 R.LCT  = 000002        SYSTAT  000000R  024 T.DIM  = 000100      ...GBL= 000000
RANDAC  003064R     025 R.LN   = 000002        SY.BAD= 000001       T.FUN  = 010000      ...TPC= 000020
RANDY   003062R     025 R.LNUM= 000010         SY.BRK= 100000       T.LIST= 000400

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

H 1

```
ASSIGN  MACRO M1113  08-FEB-80 10:48  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000164      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12939 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:00
ASSIGN,DM1:[305,301]ASSIGN/-SP=APLRX1,APLDEF,ASSIGN,END
```

I 1

```
**FILE**ID**BINSCH
```

```
BBBBBBBB        IIIIII      NN      NN     SSSSSSSS      CCCCCCCC    HH      HH
BBBBBBBB        IIIIII      NN      NN     SSSSSSSS      CCCCCCCC    HH      HH
BB      BB        II        NN      NN   SS            CC            HH      HH
BB      BB        II        NN      NN   SS            CC            HH      HH
BB      BB        II        NNNN    NN   SS            CC            HH      HH
BB      BB        II        NNNN    NN   SS            CC            HH      HH
BBBBBBBB          II        NN  NN  NN     SSSSSS      CC            HHHHHHHHHH
BBBBBBBB          II        NN  NN  NN     SSSSSS      CC            HHHHHHHHHH
BB      BB        II        NN    NNNN           SS    CC            HH      HH
BB      BB        II        NN    NNNN           SS    CC            HH      HH
BB      BB        II        NN      NN           SS    CC            HH      HH      ....
BB      BB        II        NN      NN           SS    CC            HH      HH      ....
BBBBBBBB        IIIIII      NN      NN   SSSSSSSS      CCCCCCCC    HH      HH      ....
BBBBBBBB        IIIIII      NN      NN   SSSSSSSS      CCCCCCCC    HH      HH      ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 10:50
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    BINSCH           SEQUENTIAL TABLE SEARCH
```

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

L 1

BINSCH MACRO M1113 08-FEB-80 10:50 PAGE 2
APLDEF

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

M 1

APLDEF MACRO M1113 08-FEB-80 10:50 PAGE 3
APLDEF SYMBOL TABLE

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  BINSCH
                                .IDENT  \V02.03\

                                .SBTTL  BINSCH          SEQUENTIAL TABLE SEARCH

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is used by FIXLIN to find Mneumonics
                                ;       for characters in TTY mode and overstrikes in APL
                                ;       mode. It does a sequential search to find the value
                                ;       in the character tables.
                                ;
                                ; INPUTS:
                                ;       R0      - Value to find in the table
                                ;       R1      - Beginning of the table
                                ;       R3      - End of the table+2
                                ;
                                ; OUTPUTS:
                                ;       R2      - Offset in the table (in words)
                                ;       Z-Bit   - Clear if not found
                                ;                 Set if found
                                ;--
000000                                  .PSECT  CODE

000000                          BINSCH::
000000  010102                          MOV     R1,R2                   ;Copy pointer
000002  020203                  1$:     CMP     R2,R3                   ;Finished yet?
000004  103007                          BHIS    2$                      ;Yes, not in the table
000006  020022                          CMP     R0,(R2)+                ;Is this it?
000010  001374                          BNE     1$                      ;Nope

000012  160102                          SUB     R1,R2                   ;Yes, compute offset
000014  006202                          ASR     R2                      ;    (in words)
000016  005302                          DEC     R2                      ;Cheat for offset
000020  000264                          SEZ                             ;We found it!
000022                                  RETURN                          ;Finished

000024  005002                  2$:     CLR     R2                      ;Nothing
000026  000244                          CLZ                             ;Not found
000030                                  RETURN                          ;Finished
```

```
BINSCH  MACRO M1113  08-FEB-80 10:50  PAGE 45
BINSCH           SEQUENTIAL TABLE SEARCH

      1          000001                              .END
```

BINSCH MACRO M1113 08-FEB-80 10:50 PAGE 45-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    C.BLOK= 000064    D.DECD= 000002    EIS    = 000000
AC1    =%000001          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC2    =%000002          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC3    =%000003          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC4    =%000004          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC5    =%000005          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ALLINT  003274R     025  CHRPDL= 000271    C.CZ   = 001000    D.EQ   = 000000    ERRPTR  004770R     025
ARYRET  003276R     025  CHRPLU= 000055    C.DIG  = 000040    D.EXPD= 000002    ERRO    004774R     025
ASCQUO= 000042           CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ALEN= 000004           CHRQST= 000121    C.FLEN= 000010    D.FMT  = 000004    E$CT    005056R     025
A.ARG  = 000040          CHRQUO= 000113    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.ASB  = 000200          CHRRBR= 000047    C.FRC  = 000001    D.FPUT= 000005    FA.APD= 000100
A.BITS= 000003           CHRRPR= 000042    C.IDEN= 000010    D.GE   = 000000    FA.CRE= 000010
A.CON  = 000020          CHRSEM= 000074    C.NAME= 000102    D.GT   = 000000    FA.DLK= 001000
A.DIMS= 000002           CHRSMO= 000112    C.NOBF= 000400    D.IBM  = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRSPA= 000040    C.NUM  = 000020    D.IND  = 000002    FA.EXC= 002000
A.NUM  = 000006          CHRUPA= 000131    C.OP   = 000004    D.INNR= 000013    FA.EXT= 000004
A.PTR  = 000000          CHR.BS= 000010    C.OPN  = 010000    D.INTR= 000006    FA.NSP= 000100
A.TMP  = 000010          CHR.CC= 000003    C.RDLN= 177776    D.LE   = 000000    FA.NSR= 000200
BADDIR= 104463           CHR.CR= 000015    C.REWT= 002000    D.LOG  = 000001    FA.POS= 010000
BADERR= 104437           CHR.CZ= 000032    C.RND  = 000010    D.LT   = 000000    FA.RD  = 000001
BINSCH  000000RG    020  CHR.GT= 000046    C.SBIT= 177756    D.MAX  = 000000    FA.RWD= 004000
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MEM  = 000002    FA.SEQ= 040000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MIN  = 000000    FA.SHR= 000040
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MUL  = 000000    FA.TMP= 000020
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.NE   = 000000    FA.WCK= 020000
CHAN00  000256R     025  CHR.SO= 000016    C.SRC  = 177774    D.NN   = 000000    FA.WRT= 000002
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NR   = 000000    FD.BLK= 000010
CHAN02  000512R     025  CHR.UL= 000106    C.TT   = 000004    D.OR   = 000000    FD.CCL= 000002
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FD.CR  = 000002
CHAN05  001240R     025  CHR..S= 000163    DEBUG  = 000000    D.PWR  = 000001    FD.DIR= 000010
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R     025 D.QUAD= 000000    FD.F11= 040000
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.RES  = 000000    FD.INS= 000010
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R     025 D.RHO  = 000000    FD.ISP= 002000
CHAN10  002332R     025  CHR..1= 000061    DIMAX  = 000020    D.ROT  = 000003    FD.MNT= 100000
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R     025 D.SCON= 000006    FD.OSP= 004000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006    FD.PLC= 000004
CHAN13  000022R     023  CMDMAX= 000010    DIMTYP  003116R     025 D.SUB  = 000000    FD.PRN= 000004
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFU  = 104466    D.SUBS= 000006    FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRND= 000051            COMCNT  005126R     025 DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R     025 D.UNON= 000006    FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R     025 D.AND  = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R     025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R     025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001    FF.NV  = 000003
CHRDLR= 000176           CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R     025 D.COMP= 000001    EDICHR  005063R     025 FF.RWF= 000006
CHREP  = 000105          CTMAX   000212R     025 D.CONT= 000006    EDICNT  005064R     025 FF.SPC= 000004
CHRERR= 104431           CTRMAX  000172R     025 D.CTLM= 000000    EDIFLG  005062R     025 FHEADR  005012R     025
CHRFIL= 000177           C.ALOC= 000162    D.DFAL= 000003    EDIT   = 000003    FHRET   005040R     025
```

BINSCH MACRO M1113 08-FEB-80 10:50 PAGE 45-2
SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464          MONSFT  003126R 025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540          MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565          MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025     F.FUN = 000010          INDERR= 104410          M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025     F.FVER= 000120          INNTMP  003250R 025     M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025     F.HEAD= 000006          INNTM2  005016R 025     M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025     F.HIBK= 000004          INPTR   000064R 025     M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025     F.LBL = 000020          INSNAM= 104462          M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025     F.LINE= 000012          IOERR = 104477          M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025     F.LOC = 000004          I.CNT = 177776          M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772          M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025     F.LUN = 000042          I.LEN = 000012          M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.OERR= 000062          M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770          M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774          M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766          M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406          M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R 025     M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025     F.NREC= 000030          LARGBL  003162R 025     M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R 025     M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R 025     M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025     F.PREV= 000000          LARGLE  003174R 025     M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025     F.PRNT= 100000          LARGLN  003166R 025     M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025     F.PROT= 010000          LARGPT  003164R 025     M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025     M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025     M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025     M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025     M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025     M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621          M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025     M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412          M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025     M.GDWN= 000003      ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025     M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025     M.IBM = 000003      ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025     M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025     M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025     M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020          M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025     M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541          M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025     M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025     M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025     M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025     M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012      NUMFLG  003230R 025
F.ERR = 000052          GNOME   003102R 025     MACHIN= 000106          M.RAVL= 000000      NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025     MAXCHN= 000014          M.RCIP= 000000      NUMNEG  003232R 025
F.FFBY= 000014          HASHTB  005140R 025     MONBLK  003120R 025     M.RDCT= 000011      NUMQFL  003234R 025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R 025     M.REVS= 000003      NUMRIG  003231R 025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R 025     M.ROLL= 000000      NUMWID  003226R 025
F.FNAM= 000110          ICHAN   005050R 025     MONPTR  003122R 025     M.SATA= 000001      NUMZFL  003235R 025
```

BINSCH MACRO M1113 08-FEB-80 10:50 PAGE 45-3
SYMBOL TABLE

```
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.CLR= 000040          T.LIT  = 000020
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR  = 000000          SY.ERP= 000020          T.MON  = 000010
N.DESC= 000020          RARGLE  003216R     025 R.SEQ  = 000003          SY.NOU= 000100          T.NIL  = 000004
N.DEV = 000032          RARGLN  003210R     025 R.VAR  = 000002          SY.OPN= 000002          T.NUM  = 002000
N.DEVN= 000034          RARGPT  003206R     025 R6     =%000006          SY.PMT= 001000          T.OPFN= 000001
N.DID = 000024          RARGSF  003214R     025 R7     =%000007          SY.SRT= 000400          T.OTHR= 040000
N.DVNM= 000032          RARGTY  003212R     025 SI     =%000005          SY.SVC= 000004          T.SOP  = 000002
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433           SY.SVL= 000200          T.SPOP= 000040
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R     025 S.AB   = 177776          T.TMP  = 000200
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R     025 S.ASB  = 100000          T.VAL  = 100000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R     025 S.BFHD= 000020          T.VAR  = 001000
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R     025 S.BITS= 000002          UPPER  = 177400
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200           S.CHR1= 000003          UTNAME  000344R     023
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R     025 S.FATT= 000016          VALERR= 104413
N.PPN = 000024          REENTR  003060R     025 SI.BIT= 177774           S.FDB  = 000140          VARBLE  007400R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772           S.FLB  = 040000          VARBLS  000000R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.RET= 177772           S.FNAM= 000006          VERNUM  000000R     025
OBUFLN= 001000          RESERR= 104440           SI.VAL= 177776          S.FNB  = 000036          VRSION= 000002
OCHAN   005052R     025 RESLEN  003150R     025 SKPSIZ  005134R     025 S.FNBW= 000017          WIDTH   000072R     025
OMODE   000402R     023 RESPTR  003140R     025 SPHOLD  000004R     024 S.FNTY= 000004          WROUND  000020R     025
OPTMP0  005066R     025 RESSFT  003146R     025 SPINIT  000002R     024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP1  005070R     025 RESTYP  003144R     025 STKINC= 000774           S.LEN  = 000002          WSFULL= 104403
OPTMP2  005072R     025 RESULT  003136R     025 STOP   = 104420          S.LINK= 177774          WSNAME  000074R     025
OPTMP3  005074R     025 RESVAL  003154R     025 SUBCNT  003104R     025 S.NFEN= 000020          WSTIME  000132R     025
OPTMP4  005076R     025 RINC    005104R     025 SUBDCT  003106R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SUBPTR  003110R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMB1   005344R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMF    005340R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMPT1  005346R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP9  005110R     025 RSTS   = 000000          SYMTAB  005352R     025 TMODE   000403R     023 $RSX   = ****** G
OPTM10  005112R     025 RSX    = 000115          SYMTBF= 104404           TMPTAG= 000166          $RT11  = ****** G
OPTM14  005026R     025 RTOP    005070R     025 SYMTOP  005350R     025 TTYBUF  004140R     025 .BOOLE= 000002
ORIGIN  000030R     025 RT11   = 000000          SYMTP   006272R     025 TTYLEN= 000622           .CHARA= 000000
OUTPTR  000066R     025 R.BITS= 000003           SYMWRN= 000054           TURKEY= 104436          .DEF.  = 000002
OVROPR= 000000          R.FIX  = 000001          SYNERR= 104407           TYPE   = 000166          .FLOAT= 000006
PATCH   007344R     025 R.LCT  = 000002          SYSERR= 104402           T.BOL  = 020000          .FSRCB= ****** G
PRECSN  000052R     025 R.LN   = 000002          SYSTAT  000000R     024 T.CON  = 004000          .INTEG= 000004
RANDAC  003064R     025 R.LNUM= 000010           SY.BAD= 000001           T.DIM  = 000100          .MOLUN  000404R     023
RANDY   003062R     025 R.NEXT= 000006           SY.BRK= 100000           T.FUN  = 010000          ...GBL= 000000
RANERR= 104411          R.OFF  = 000000          SY.BS  = 000010          T.LIST= 000400           ...TPC= 000020
RARGBL  003204R     025

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

```
BINSCH  MACRO M1113  08-FEB-80 10:50  PAGE 45-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000032      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12904 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:56
BINSCH,DM1:[305,301]BINSCH/-SP=APLRX1,APLDEF,BINSCH,END
```

```
**FILE**ID**BLDLST
```

```
BBBBBBBB     LL           DDDDDDDD     LL             SSSSSSSS   TTTTTTTTTT
BBBBBBBB     LL           DDDDDDDD     LL             SSSSSSSS   TTTTTTTTTT
BB      BB   LL           DD      DD   LL           SS             TT
BB      BB   LL           DD      DD   LL           SS             TT
BB      BB   LL           DD      DD   LL           SS             TT
BB      BB   LL           DD      DD   LL           SS             TT
BBBBBBBB     LL           DD      DD   LL             SSSSSS       TT
BBBBBBBB     LL           DD      DD   LL             SSSSSS       TT
BB      BB   LL           DD      DD   LL                   SS     TT
BB      BB   LL           DD      DD   LL                   SS     TT
BB      BB   LL           DD      DD   LL                   SS     TT          ....
BB      BB   LL           DD      DD   LL                   SS     TT          ....
BBBBBBBB     LLLLLLLLLL   DDDDDDDD     LLLLLLLLLL   SSSSSSSS       TT          ....
BBBBBBBB     LLLLLLLLLL   DDDDDDDD     LLLLLLLLLL   SSSSSSSS       TT          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS             TT
LL           SS             TT
LL           SS             TT
LL           SS             TT
LL             SSSSSS       TT
LL             SSSSSS       TT
LL                   SS     TT
LL                   SS     TT
LL                   SS     TT
LL                   SS     TT
LLLLLLLLLL   SSSSSSSS       TT
LLLLLLLLLL   SSSSSSSS       TT
```

.MAIN. MACRO M1113 08-FEB-80 10:52
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    BLDLST           BUILD A LIST FROM SYMBOLS
```

BLDLST MACRO M1113 08-FEB-80 10:52 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'  ISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  BLDLST
                                .IDENT  \V02.03\

                                .SBTTL  BLDLST          BUILD A LIST FROM SYMBOLS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is used to take a command line and
                                ;       extract a list of symbol names from it, placing
                                ;       the names into a list block (i.e., each entry
                                ;       in the list block points to the entry in the
                                ;       symbol table)
                                ;
                                ; INPUTS:
                                ;       R2      - The next character to handle
                                ;       R3      - Count of characters remaining
                                ;       R4      - The pointer to the next character
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       The list block is placed on the top of the SI.
                                ;
                                ; ROUTINES CALLED:
                                ;       GETCOR  - Get core for the list block
                                ;       GETNB2  - Get a non-blank character
                                ;       SYMBOL  - Enter/Find a symbol in the symbol table
                                ;--
000000                                  .PSECT  CODE

000000                          BLDLST::
000000                                  PUSH    <R0,R1,R2,R3,R4>        ;He makes his saving roll
000012  005046                          CLR     -(SP)                   ;Init the count
000014                          1$:     CALL    GETNB2                  ;Get a non-blank character
000020  002423                          BLT     4$                      ;Nothing left
000022  120227  000015                  CMPB    R2,#CHR.CR              ;A <CR>?
000026  001420                          BEQ     4$                      ;Yes, EOL
000030  132762  000010  000000G        BITB    #C.IDEN,CHRTAB(R2)      ;An Alphabetic?
000036  001413                          BEQ     3$                      ;No, SYNTAX ERROR
000040  005216                          INC     (SP)                    ;Increase count of elements
000042                          2$:     GETC    4$                      ;Get the next character
000042  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000044  002411                          BLT 4$                  ;NO, BRANCH TO 4$
000046  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000050  042702  177400                  BIC     #<UPPER>,R2             ;Clear out garbage
000054  132762  000050  000000G        BITB    #C.IDEN!C.DIG,CHRTAB(R2);A Symbol Character?
000062  001367                          BNE     2$                      ;Yes, continue
000064  000753                          BR      1$                      ;Loop for each name

000066  104407                  3$:     SYNERR                          ;Syntax Error

000070                          4$:     POP     R1                      ;Get the count
000072  001775                          BEQ     3$                      ;Must be at least 1
000074  010100                          MOV     R1,R0                   ;Copy the count
000076  006300                          ASL     R0                      ;Count in bytes
000100  062700  000006                  ADD     #L.PTR1+2.,R0           ;Add in overhead
000104                                  CALL    GETCOR                  ;Get some memory
000110  012725  000000G                 MOV     #SYSER,(SI)+            ;Push return address on the SI
000114  012725  000400                  MOV     #T.LIST,(SI)+           ;It's a list
```

```
BLDLST  MACRO M1113  08-FEB-80 10:52  PAGE 44-1
BLDLST          BUILD A LIST FROM SYMBOLS

     55 000120  010025                          MOV     R0,(SI)+            ;Point to the list block
     56 000122  010010                          MOV     R0,(R0)             ;Point to itself
     57 000124  062720  000006                  ADD     #L.PTR1+2.,(R0)+    ;Point to first element
     58 000130  110120                          MOVB    R1,(R0)+            ;Get # of elements
     59 000132  105020                          CLRB    (R0)+               ;No flags in high byte
     60 000134  010010                          MOV     R0,(R0)             ;Last element
     61 000136  062720  000002                  ADD     #2.,(R0)+           ;   pointer
     62
     63 000142                                  POP     <R4,R3,R2>          ;Restore the data
     64 000150                                  PUSH    R1                  ;Save the count
     65 000152  010001                          MOV     R0,R1               ;Copy data pointer
     66 000154                          5$:     PUSH    R1                  ;Save data pointer
     67 000156                                  CALL    GETNB2              ;Get Non-Blank Element
     68 000162                                  CALL    SYMBOL              ;Get the next symbol
     69 000166                                  POP     R1                  ;Restore list pointer
     70 000170  010021                          MOV     R0,(R1)+            ;Copy the symbol entry address
     71 000172  005316                          DEC     (SP)                ;Decrement loop count
     72 000174  003367                          BGT     5$                  ;   and loop until fini'
     73
     74 000176  005726                          TST     (SP)+               ;Delete Count from the stack
     75 000200                                  POP     <R1,R0>             ;Restore registers
     76 000204                                  RETURN                      ;Finished
```



```
BLDLST  MACRO M1113  08-FEB-80 10:52  PAGE 45
```

```
BLDLST  MACRO M1113  08-FEB-80 10:52  PAGE 45
BLDLST          BUILD A LIST FROM SYMBOLS

     1          000001                          .END
```

```
BLDLST  MACRO M1113  08-FEB-80 10:52  PAGE 45-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135          C.ALOC= 000162          D.DEAL= 000003          EDIT  = 000003
AC1    =%000001          CHRLBR= 000073          C.BLOK= 000064          D.DECD= 000002          EIS   = 000000
AC2    =%000002          CHRLPR= 000072          C.BLSZ= 000140          D.DIV = 000000          ELESIZ  005132R     025
AC3    =%000003          CHRMNS= 000137          C.BUFF= 177760          D.DMD = 000010          ENDCOR  000004R     025
AC4    =%000004          CHRMUL= 000075          C.BUFL= 177762          D.DQAD= 177777          EOF   = 104475
AC5    =%000005          CHRNEG= 000100          C.CCNT= 177766          D.DROP= 000002          ERRMAX= 000166
ALLINT  003274R     025  CHRPDL= 000271          C.CPTR= 177764          D.ENCD= 000002          ERRNUM  004772R     025
ARYRET  003276R     025  CHRPLU= 000055          C.CZ  = 001000          D.EQ  = 000000          ERRPTR  004770R     025
ASCQUO= 000042           CHRPWR= 000120          C.DIG = 000040          D.EXPD= 000002          ERR0    004774R     025
A.ALEN= 000004           CHRQST= 000121          C.FASW= 020000          D.FGET= 000005          EXEBUF  005054R     025
A.ARG = 000040           CHRQUO= 000113          C.FLEN= 000010          D.FMT = 000004          E$CT    005056R     025
A.ASB = 000200           CHRRBR= 000047          C.FLSH= 004000          D.FPNT= 000005          E$PT    005060R     025
A.BITS= 000003           CHRRPR= 000042          C.FRC = 000001          D.FPUT= 000005          FA.APD= 000100
A.CON = 000020           CHRSEM= 000074          C.IDEN= 000010          D.GE  = 000000          FA.CRE= 000010
A.DIMS= 000002           CHRSMO= 000112          C.NAME= 000102          D.GT  = 000000          FA.DLK= 001000
A.DIM1= 000006           CHRSPA= 000040          C.NOBF= 000400          D.IBM = 000002          FA.ENB= 100000
A.NUM = 000006           CHRTAB= ****** GX       C.NUM = 000020          D.IND = 000002          FA.EXC= 002000
A.PTR = 000000           CHRUPA= 000131          C.OP  = 000004          D.INNR= 000013          FA.EXT= 000004
A.TMP = 000010           CHR.BS= 000010          C.OPN = 010000          D.INTR= 000006          FA.NSP= 000100
BADDIR= 104463           CHR.CC= 000003          C.RDLN= 177776          D.LE  = 000000          FA.NSR= 000200
BADERR= 104437           CHR.CR= 000015          C.REWT= 002000          D.LOG = 000001          FA.POS= 010000
BLDLST  000000RG    020  CHR.CZ= 000032          C.RND = 000010          D.LT  = 000000          FA.RD = 000001
BLZBUB  005116R     025  CHR.GT= 000046          C.SBIT= 177756          D.MAX = 000000          FA.RWD= 004000
BRSELF  005046R     025  CHR.LF= 000012          C.SPFX= 000100          D.MEM = 000002          FA.SEQ= 040000
CCNT    005072R     025  CHR.LT= 000043          C.SPGL= 000200          D.MIN = 000000          FA.SHR= 000040
CHANEX  003060R     025  CHR.SI= 000017          C.SPLX= 000001          D.MUL = 000000          FA.TMP= 000020
CHAN00  000256R     025  CHR.SL= 000057          C.SPRS= 000002          D.NE  = 000000          FA.WCK= 020000
CHAN01  000330R     025  CHR.SO= 000016          C.SRC = 177774          D.NN  = 000000          FA.WRT= 000002
CHAN02  000512R     025  CHR.TB= 000011          C.SRCP= 000024          D.NR  = 000000          FD.BLK= 000010
CHAN03  000674R     025  CHR.UL= 000106          C.TT  = 000004          D.OR  = 000000          FD.CCL= 000002
CHAN04  001056R     025  CHR..A= 000141          C.WIDC= 177770          D.PLUS= 000000          FD.COM= 020000
CHAN05  001240R     025  CHR..E= 000145          C.WIDR= 177772          D.PSUB= 000006          FD.CR = 000002
CHAN06  001422R     025  CHR..S= 000163          DEBUG = 000000          D.PWR = 000001          FD.DIR= 000010
CHAN07  001604R     025  CHR..T= 000164          DEFERR= 104405          D.QQAD= 000000          FD.FTN= 000001
CHAN08  001766R     025  CHR..U= 000165          DEVIL   005114R     025 D.QUAD= 000000          FD.F11= 040000
CHAN09  002150R     025  CHR..Z= 000172          DEVNTF= 104470          D.RES = 000000          FD.INS= 000010
CHAN10  002332R     025  CHR..0= 000060          DIGITS  000006R     025 D.RHO = 000000          FD.ISP= 002000
CHAN11  002514R     025  CHR..1= 000061          DIMAX = 000020          D.ROT = 000003          FD.MNT= 100000
CHAN12  002676R     025  CH.AND= 000001          DIMBLK  003112R     025 D.SCON= 000006          FD.OSP= 004000
CHAN13  000022R     023  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006          FD.PLC= 000004
CHAN14  000204R     023  CMDMAX= 000010          DIMTYP  003116R     025 D.SUB = 000000          FD.PRN= 000004
CHNOPN= 104471           CNTSIZ  005130R     025 DIRFUL= 104466          D.SUBS= 000006          FD.PSE= 010000
CHRAND= 000051           COMBIN  005076R     025 DOMERR= 104417          D.TAKE= 000002          FD.RAH= 000001
CHRBEL= 000007           COMCNT  005126R     025 DPTERR= 104416          D.TRAN= 000004          FD.RAN= 000002
CHRBKR= 000133           COMERR= 104425          DYDCNT  003224R     025 D.UNON= 000006          FD.REC= 000001
CHRBOL= 000200           COMPTR  005122R     025 D.AND = 000000          D.$MAX= 000000          FD.RWM= 000001
CHRBOX= 000114           COMRTN  003156R     025 D.BXCT= 000007          D.$MIN= 000000          FD.SDI= 000020
CHRCMA= 000054           COMSKP  005124R     025 D.BXIO= 000007          D.$MUL= 000000          FD.SQD= 000040
CHRCOL= 000076           CORPTR  005120R     025 D.BXNL= 000007          D.$PLU= 000000          FD.TTY= 000004
CHRDEL= 000107           CS.ASS= 000200          D.BXPP= 000007          D.$RES= 000000          FD.WBH= 000002
CHRDIV= 000053           CS.BOO= 000010          D.BXPW= 000007          D.$SUB= 000000          FF.CHR= 000005
CHRDLR= 000176           CS.FLO= 000020          D.BXRL= 000007          D.1COM= 000001          FF.NV = 000003
CHRDLT= 000110           CS.IDN= 000040          D.CIRC= 000001          D.1EXP= 000002          FF.POE= 000002
CHRDOT= 000056           CS.ONE= 000100          D.COMB= 000001          D.1ROT= 000003          FF.RWD= 000001
CHREP = 000105           CTEPS   000152R     025 D.COMP= 000001          EDICHR  005063R     025 FF.RWF= 000006
CHRERR= 104431           CTMAX   000212R     025 D.CONT= 000006          EDICNT  005064R     025 FF.SPC= 000004
CHRFIL= 000177           CTRMAX  000172R     025 D.CTLM= 000000          EDIFLG  005062R     025 FHEADR  005012R     025
```

SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110      HSHMSK= 177601          MONCVT  003132R     025 M.REVS= 000003
FILNTF= 104467          F.FNB = 000102      HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000
FIS   = 000000          F.FNDF= 000010      ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
FISCON= 046000          F.FTYP= 000116      ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FIXPTR  003074R     025 F.FUN = 000010      ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FLAGS   004764R     025 F.FVER= 000120      ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FLCBUF  003250R     025 F.HEAD= 000006      INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLINE   005014R     025 F.HIBK= 000004      INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
FLOEXP  003270R     025 F.LBL = 000020      INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
FLONEG  003272R     025 F.LINE= 000012      INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
FLSWTB  000222R     025 F.LOC = 000004      INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLTLEN= 000010          F.LSTC= 000004      IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FNINSI  005044R     025 F.LUN = 000042      I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FO.APD= 000106          F.MBC1= 000055      I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FO.MFY= 000002          F.MBFG= 000056      I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.RD = 000001          F.MON = 000002      I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.UPD= 000006          F.NIL = 000001      I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.WRT= 000016          F.NRBD= 000024      I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FPMODE  000042R     025 F.NREC= 000030      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FPP   = 000000          F.OPN = 000010      LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FPPLEN= 000004          F.OVBS= 000030      LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FREBLK  007400R     025 F.PREV= 000000      LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FUZZ    000142R     025 F.PRNT= 100000      LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
FXPTR1  003076R     025 F.PROT= 010000      LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
F$EVAL= 000100          F.RACC= 000016      LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
F$EVI = 000200          F.RATT= 000001      LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$EXEC= 000400          F.RCNM= 000034      LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F$FUNC= 001000          F.RCTL= 000017      LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F.ACTL= 000076          F.READ= 004000      LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F.ALOC= 000040          F.REDF= 000040      LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.BBFS= 000062          F.RES = 000040      LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.BDB = 000070          F.RET = 177772      LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002      LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BITS= 177774          F.RTYP= 000000      LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BKDN= 000026          F.R2  = 000014      LENTAB  000054R     025 M.GDWN= 000003          ND.$ES= 000020
F.BKDS= 000020          F.R3  = 000016      LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKEF= 000050          F.R4  = 000020      LFTOP   005066R     025 M.IBM = 000003          ND.$NS= 000040
F.BKP1= 000051          F.SEQN= 000100      LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKST= 000024          F.SPDV= 000072      LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BKVB= 000064          F.SPUN= 000074      LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.BR  = 000020          F.STBK= 000036      LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.CHR = 000075          F.STK = 000022      LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.CNTG= 000034          F.STOP= 000004      LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CODE= 000010          F.STP = 000001      LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.DFNB= 000046          F.SUSP= 000002      LOWCOR  000002R     025 M.NOT = 000000          NOTAVL=  04472
F.DSPT= 000044          F.TRC = 000002      LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.DVSP= 000134          F.UNIT= 000136      LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.EFBK= 000010          F.URBD= 000020      L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.EFN = 000050          F.VAL = 177776      L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.END = 002000          F.VBN = 000064      L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EOBB= 000032          F.VBSZ= 000060      L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.ERR = 000052          GETCOR= ****** GX   L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.FACC= 000043          GETNB2= ****** GX   L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.FFBY= 000014          GNOME   003102R     025 MACHIN= 000106      M.RAVL= 000000          NUMLEN  000062R     025
F.FLAG= 000002          HASHC   005136R     025 MAXCHN= 000014      M.RCIP= 000000          NUMNEG  003232R     025
F.FLB = 000100          HASHTB  005140R     025 MONBLK  003120R     025 M.RDCT= 000011      NUMQFL  003234R     025
```

BLDLST MACRO M1113 08-FEB-80 10:52 PAGE 45-3
SYMBOL TABLE

```
NUMRIG  003231R     025 RANERR= 104411          R.OFF  = 000000          SY.BRK= 100000          T.LIST= 000400
NUMWID  003226R     025 RARGBL  003204R   025 R.PREV= 000004          SY.BS = 000010          T.LIT = 000020
NUMZFL  003235R     025 RARGCV  003220R   025 R.PTR  = 000000          SY.CLR= 000040          T.MON = 000010
NYIERR= 104432          RARGFL  003202R   025 R.SEQ  = 000003          SY.ERR= 000020          T.NIL = 000004
N.BLSZ= 000036          RARGLE  003216R   025 R.VAR  = 000002          SY.NOU= 000100          T.NUM = 002000
N.DESC= 000020          RARGLN  003210R   025 R6     =%000006          SY.OPN= 000002          T.OPFN= 000001
N.DEV = 000032          RARGPT  003206R   025 R7     =%000007          SY.PMT= 001000          T.OTHR= 040000
N.DEVN= 000034          RARGSF  003214R   025 SI     =%000005          SY.SRT= 000400          T.SOP = 000002
N.DID = 000024          RARGTY  003212R   025 SIDAMN= 104433           SY.SVC= 000004          T.SPOP= 000040
N.DVNM= 000032          RARGVA  003222R   025 SIFRAM  003314R   025 SY.SVL= 000200          T.TMP = 000200
N.EXT = 000014          RBASE   005074R   025 SISTAK  006350R   025 S.AB  = 177776          T.VAL = 100000
N.FID = 000000          RCOUNR  005112R   025 SITOP   006346R   025 S.ASB = 100000          T.VAR = 001000
N.FNAM= 000006          RCOUNT  005110R   025 SITP    007144R   025 S.BFHD= 000020          UPPER = 177400
N.FTYP= 000014          REDCNT  005070R   025 SIWARN= 000200           S.BITS= 000002          UTNAME  000344R     023
N.FVER= 000016          REDIGN  005066R   025 SIZTAB  000044R   025 S.CHR1= 000003          VALERR= 104413
N.NAME= 000006          REDOFF  005072R   025 SI.BIT= 177774           S.FATT= 000016          VARBLE  007400R     025
N.NEXT= 000022          REENTR  003060R   025 SI.LEN= 177772           S.FDB = 000140          VARBLS  000000R     025
N.PPN = 000024          RESBIT  003142R   025 SI.RET= 177772           S.FLB = 040000          VERNUM  000000R     025
N.STAT= 000020          RESCVT  003152R   025 SI.VAL= 177776           S.FNAM= 000006          VRSION= 000002
N.UNIT= 000034          RESERR= 104440          SKPSIZ  005134R   025 S.FNB = 000036          WIDTH   000072R     025
OBUFLN= 001000          RESLEN  003150R   025 SPHOLD  000004R   024 S.FNBW= 000017          WROUND  000020R     025
OCHAN   005052R     025 RESPTR  003140R   025 SPINIT  000002R   024 S.FNTY= 000004          WSDATE  000136R     025
OMODE   000402R     023 RESSFT  003146R   025 STKINC= 000774           S.FTYP= 000002          WSFULL= 104403
OPTMP0  005066R     025 RESTYP  003144R   025 STOP   = 104420          S.LEN = 000002          WSNAME  000074R     025
OPTMP1  005070R     025 RESULT  003136R   025 SUBCNT  003104R   025 S.LINK= 177774          WSTIME  000132R     025
OPTMP2  005072R     025 RESVAL  003154R   025 SUBDCT  003106R   025 S.NFEN= 000020          $FIS   = ****** G
OPTMP3  005074R     025 RINC    005104R   025 SUBPTR  003110R   025 S.TYPE= 140000          $FPP   = ****** G
OPTMP4  005076R     025 RLTBAS  005066R   025 SYMBOL= ****** GX       S.VAL = 000000          $FPPLN= ****** G
OPTMP5  005100R     025 ROTINC  005070R   025 SYMB1   005344R   025 TKDIMS  005072R   025 $MACHN= ****** G
OPTMP6  005102R     025 ROUND   000010R   025 SYMF    005340R   025 TKFRAM  005066R   025 $RSTS  = ****** G
OPTMP7  005104R     025 ROVER   003312R   025 SYMPT1  005346R   025 TKLEFT  005070R   025 $RSX   = ****** G
OPTMP8  005106R     025 RSTS   = 000000         SYMTAB  005352R   025 TMODE   000403R   023 $RT11  = ****** G
OPTMP9  005110R     025 RSX    = 000115         SYMTBF= 104404          TMPTAG= 000166          .BOOLE= 000002
OPTM10  005112R     025 RTOP    005070R   025 SYMTOP  005350R   025 TTYBUF  004140R   025 .CHARA= 000000
OPTM14  005026R     025 RT11   = 000000         SYMTP   006272R   025 TTYLEN= 000622          .DEF. = 000002
ORIGIN  000030R     025 R.BITS= 000003          SYMWRN= 000054          TURKEY= 104436          .FLOAT= 000006
OUTPTR  000066R     025 R.FIX  = 000001         SYNERR= 104407          TYPE   = 000166         .FSRCB= ****** G
OVROPR= 000000          R.LCT  = 000002         SYSER  = ****** GX      T.BOL  = 020000         .INTEG= 000004
PATCH   007344R     025 R.LN   = 000002         SYSERR= 104402          T.CON  = 004000         .MOLUN  000404R     023
PRECSN  000052R     025 R.LNUM= 000010          SYSTAT  000000R   024 T.DIM  = 000100         ...GBL= 000000
RANDAC  003064R     025 R.NEXT= 000006          SY.BAD= 000001          T.FUN  = 010000         ...TPC= 000020
RANDY   003062R     025

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

```
BLDLST  MACRO M1113  08-FEB-80 10:52  PAGE 45-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000206      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12938 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:02
BLDLST,DM1:[305,301]BLDLST/-SP=APLRX1,APLDEF,BLDLST,END
```

```
**FILE**ID**BRKCHK
```

```
BBBBBBBB      RRRRRRRR      KK       KK      CCCCCCCC    HH       HH    KK       KK
BBBBBBBB      RRRRRRRR      KK       KK      CCCCCCCC    HH       HH    KK       KK
BB       BB   RR       RR   KK       KK    CC            HH       HH    KK       KK
BB       BB   RR       RR   KK       KK    CC            HH       HH    KK       KK
BB       BB   RR       RR   KK     KK      CC            HH       HH    KK     KK
BB       BB   RR       RR   KK     KK      CC            HH       HH    KK     KK
BBBBBBBB      RRRRRRRR      KKKKKK         CC            HHHHHHHHHH    KKKKKK
BBBBBBBB      RRRRRRRR      KKKKKK         CC            HHHHHHHHHH    KKKKKK
BB       BB   RR   RR       KK     KK      CC            HH       HH    KK     KK
BB       BB   RR   RR       KK     KK      CC            HH       HH    KK     KK
BB       BB   RR     RR     KK       KK    CC            HH       HH    KK       KK        ....
BB       BB   RR     RR     KK       KK    CC            HH       HH    KK       KK        ....
BBBBBBBB      RR       RR   KK       KK      CCCCCCCC    HH       HH    KK       KK        ....
BBBBBBBB      RR       RR   KK       KK      CCCCCCCC    HH       HH    KK       KK        ....
```

```
LL                SSSSSSSS    TTTTTTTTTT
LL                SSSSSSSS    TTTTTTTTTT
LL              SS                TT
LL              SS                TT
LL              SS                TT
LL              SS                TT
LL                SSSSSS          TT
LL                SSSSSS          TT
LL                      SS        TT
LL                      SS        TT
LL                      SS        TT
LL                      SS        TT
LLLLLLLLLL    SSSSSSSS            TT
LLLLLLLLLL    SSSSSSSS            TT
```

.MAIN. MACRO M1113 08-FEB-80 10:54
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    BRKCHK           CHECK TO SEE IF THE USER WANTS TO STOP
```

BRKCHK MACRO M1113 08-FEB-80 10:54
TABLE OF CONTENTS

44-	26	CCTRAP	AST FOR UNSOLICITED CHARACTERS
44-	42	RSXERR	CHECK FOR RSX ERRORS
44-	64	FCSERR	CHECK FOR FCS ERRORS

BRKCHK MACRO M1113 08-FEB-80 10:54 PAGE 1

```
                                        ;
                                        ;
                                        ;
```

J 3


```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

K 3

```
APL     MACRO M1113  08-FEB-80 10:54  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  BRKCHK
                                .IDENT  \V02.03\

                                .SBTTL  BRKCHK          CHECK TO SEE IF THE USER WANTS TO STOP

000000                                  .PSECT  CODE

000000                          BRKCHK::
000000  032737  100000  000000' BIT     #SY.BRK,SYSTAT  ;DOES HE?
000006  001001                  BNE     1$              ;YES
000010                          RETURN                  ;NO, QUICKLY RETURN
000012  042737  100000  000000' 1$:     BIC     #SY.BRK,SYSTAT  ;YES, CLEAR BREAK FLAG
000020  104420                  STOP                    ;*** EXECUTION STOP ***


                                .SBTTL  CCTRAP          AST FOR UNSOLICITED CHARACTERS

000022                                  .PSECT  CODE

000022                          CCTRAP::
000022  122726  000003          CMPB    #CHR.CC,(SP)+   ;WAS IT A ^C?
000026  001012                  BNE     2$              ;NOPE
000030  032737  100000  000000' BIT     #SY.BRK,SYSTAT  ;ALREADY TRYING TO STOP?
000036  001403                  BEQ     1$              ;NO, PLEASE CONTINUE
000040  012766  000000G 000002  MOV     #DIE,2.(SP)     ;RETURN ADDRESS IS DEATH
000046  052737  100000  000000' 1$:     BIS     #SY.BRK,SYSTAT  ;INDICATE STOP REQUESTED
000054                          2$:     ASTX$S                  ;FINISHED, NOW RETURN
000054  012746                  MOV     (PC)+,-(SP)
000056     163     001          .BYTE   115.,1
000060  104377                  EMT     ^O<377>

                                .SBTTL  RSXERR          CHECK FOR RSX ERRORS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine checks the carry bit. If set, an error
                                ;       has occurred and we need to get the error code from
                                ;       the directive status word ($DSW). The routine  then
                                ;       branches to ERROR.
                                ;--

000062                                  .PSECT  CODE

000062                          RSXERR::
000062  103401                  BCS     2$              ;Yes
000064                          1$:     RETURN                  ;No, still good

000066  005737  000000G 2$:     TST     $DSW            ;Test error number
000072  002374                  BGE     1$              ;Nothing really the matter
000074                          PUSH    $DSW            ;Put the error # on the stack
000100  000137  000000G         JMP     ERROR           ;    and go


                                .SBTTL  FCSERR          CHECK FOR FCS ERRORS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
```

```
                        ;       This routine checks the carry bit. If set, an error
                        ;       has occurred and we need to get the error code from
                        ;       the FDB (in offset F.ERR). The routine then branches
                        ;       to ERROR.
                        ;--

000104                          .PSECT  CODE

000104                  FCSERR::
000104  103401                 BCS     2$              ;Yes
000106                  1$:     RETURN                  ;No, still good

000110  105760  000052  2$:     TSTB    F.ERR(R0)       ;Test error number
000114  002374                 BGE     1$              ;Nothing really the matter
000116                          PUSH    F.ERR(R0)       ;Put the error # on the stack
000122  105366  000001         DECB    1.(SP)          ;Hack conversion
000126  000137  000000G        JMP     ERROR           ;    and go
```

FCSERR CHECK FOR FCS ERRORS

```
      1          000001                              .END
```

```
BRKCHK  MACRO M1113  08-FEB-80 10:54  PAGE 47-1
SYMBOL TABLE

AC0    =%000000          CHRFIL= 000177    C.ALOC= 000162    D.CTLM= 000000    EDIFLG  005062R     025
AC1    =%000001          CHRFRO= 000135    C.BLOK= 000064    D.DEAL= 000003    EDIT   = 000003
AC2    =%000002          CHRLBR= 000073    C.BLSZ= 000140    D.DECD= 000002    EIS    = 000000
AC3    =%000003          CHRLPR= 000072    C.BUFF= 177760    D.DIV = 000000    ELESIZ  005132R     025
AC4    =%000004          CHRMNS= 000137    C.BUFL= 177762    D.DMD = 000010    ENDCOR  000004R     025
AC5    =%000005          CHRMUL= 000075    C.CCNT= 177766    D.DQAD= 177777    EOF    = 104475
ALLINT  003274R     025  CHRNEG= 000100    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
ARYRET  003276R     025  CHRPDL= 000271    C.CZ   = 001000   D.ENCD= 000002    ERRNUM  004772R     025
ASCQUO= 000042           CHRPLU= 000055    C.DIG  = 000040   D.EQ   = 000000    ERROR = ****** GX
A.ALEN= 000004           CHRPWR= 000120    C.FASW= 020000    D.EXPD= 000002    ERRPTR  004770R     025
A.ARG  = 000040          CHRQST= 000121    C.FLEN= 000010    D.FGET= 000005    ERRO    004774R     025
A.ASB  = 000200          CHRQUO= 000113    C.FLSH= 004000    D.FMT  = 000004   EXEBUF  005054R     025
A.BITS= 000003           CHRRBR= 000047    C.FRC  = 000001   D.FPNT= 000005    E$CT    005056R     025
A.CON  = 000020          CHRRPR= 000042    C.IDEN= 000010    D.FPUT= 000005    E$PT    005060R     025
A.DIMS= 000002           CHRSEM= 000074    C.NAME= 000102    D.GE   = 000000   FA.APD= 000100
A.DIM1= 000006           CHRSMO= 000112    C.NOBF= 000400    D.GT   = 000000   FA.CRE= 000010
A.NUM  = 000006          CHRSPA= 000040    C.NUM  = 000020   D.IBM  = 000002   FA.DLK= 001000
A.PTR  = 000000          CHRUPA= 000131    C.OP   = 000004   D.IND  = 000002   FA.ENB= 100000
A.TMP  = 000010          CHR.BS= 000010    C.OPN  = 010000   D.INNR= 000013    FA.EXC= 002000
BADDIR= 104463           CHR.CC= 000003    C.RDLN= 177776    D.INTR= 000006    FA.EXT= 000004
BADERR= 104437           CHR.CR= 000015    C.REWT= 002000    D.LE   = 000000   FA.NSP= 000100
BLZBUB  005116R     025  CHR.CZ= 000032    C.RND  = 000010   D.LOG  = 000001   FA.NSR= 000200
BRKCHK  000000RG    020  CHR.GT= 000046    C.SBIT= 177756    D.LT   = 000000   FA.POS= 010000
BRSELF  005046R     025  CHR.LF= 000012    C.SPFX= 000100    D.MAX  = 000000   FA.RD  = 000001
CCNT    005072R     025  CHR.LT= 000043    C.SPGL= 000200    D.MEM  = 000002   FA.RWD= 004000
CCTRAP  000022RG    020  CHR.SI= 000017    C.SPLX= 000001    D.MIN  = 000000   FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.MUL  = 000000   FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016    C.SRC  = 177774   D.NE   = 000000   FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NN   = 000000   FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106    C.TT   = 000004   D.NR   = 000000   FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.OR   = 000000   FCSERR  000104RG    020
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PLUS= 000000    FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163    DEBUG  = 000000   D.PSUB= 000006    FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.PWR  = 000001   FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R     025 D.QQAD= 000000   FD.CR  = 000002
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.QUAD= 000000    FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060    DIE    = ****** GX D.RES  = 000000   FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R     025 D.RHO  = 000000   FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001    DIMAX  = 000020   D.ROT  = 000003    FD.INS= 000010
CHAN12  002676R     025  CMDBUF  003250R     025 DIMBLK  003112R     025 D.SCON= 000006   FD.ISP= 002000
CHAN13  000022R     023  CMDMAX= 000010    DIMPTR  003114R     025 D.SDIF= 000006   FD.MNT= 100000
CHAN14  000204R     023  CNTSIZ  005130R     02  DIMTYP  003116R     025 D.SUB  = 000000   FD.OSP= 004000
CHNOPN= 104471           COMBIN  005076R     025 DIRFUL= 104466    D.SUBS= 000006   FD.PLC= 000004
CHRAND= 000051           COMCNT  005126R     025 DOMERR= 104417    D.TAKE= 000002   FD.PRN= 000004
CHRBEL= 000007           COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.PSE= 010000
CHRBKR= 000133           COMPTR  005122R     025 DYDCNT  003224R     025 D.UNON= 000006   FD.RAH= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.AND  = 000000   D.$MAX= 000000   FD.RAN= 000002
CHRBOX= 000114           COMSKP  005124R     025 D.BXCT= 000007    D.$MIN= 000000   FD.REC= 000001
CHRCMA= 000054           CORPTR  005120R     025 D.BXIO= 000007    D.$MUL= 000000   FD.RWM= 000001
CHRCOL= 000076           CS.ASS= 000200    D.BXNL= 000007    D.$PLU= 000000   FD.SDI= 000020
CHRDEL= 000107           CS.BOO= 000010    D.BXPP= 000007    D.$RES= 000000   FD.SQD= 000040
CHRDIV= 000053           CS.FLO= 000020    D.BXPW= 000007    D.$SUB= 000000   FD.TTY= 000004
CHRDLR= 000176           CS.IDN= 000040    D.BXRL= 000007    D.1COM= 000001   FD.WBH= 000002
CHRDLT= 000110           CS.ONE= 000100    D.CIRC= 000001    D.1EXP= 000002   FF.CHR= 000005
CHRDOT= 000056           CTEPS   000152R     025 D.COMB= 000001    D.1ROT= 000003   FF.NV  = 000003
CHREP  = 000105          CTMAX   000212R     025 D.COMP= 000001    EDICHR  005063R     025 FF.POE= 000002
CHRERR= 104431           CTRMAX  000172R     025 D.CONT= 000006    EDICNT  005064R     025 FF.RWD= 000001
```

BRKCHK MACRO M1113 08-FEB-80 10:54 PAGE 47-2
SYMBOL TABLE

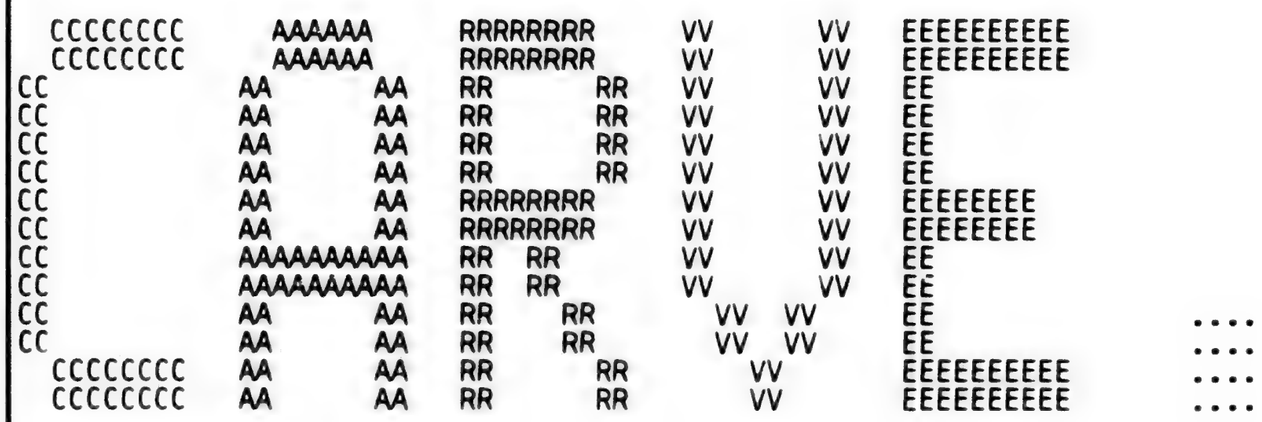
```
FF.RWF= C00006          F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011
FF.SPC= 000004          F.FLAG= 000002          HSHMSK= 177601      MONCVT  003132R 025 M.REVS= 000003
FHEADR  005012R 025     F.FLB = 000100          HSHSIZ= 000200      MONLEN  003130R 025 M.ROLL= 000000
FHRET   005040R 025     F.FNAM= 000110          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025     F.FUN = 000010          INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025     F.FVER= 000120          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025     F.HEAD= 000006          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025     F.HIBK= 000004          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025     F.LBL = 000020          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025     F.LINE= 000012          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025     F.LOC = 000004          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025     F.LUN = 000042          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025     F.NREC= 000030          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025     F.PREV= 000000          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025     F.PRNT= 100000          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025     F.PROT= 010000          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025 M.GDWN= 000003      ND.$ES= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025 M.IBM = 000003      ND.$NS= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
F.ERR = 000052          GNOME   003102R 025     MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025     MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
```

BRKCHK MACRO M1113 08-FEB-80 10:54 PAGE 47-3
SYMBOL TABLE

```
NUMQFL  003234R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BS  = 000010          T.LIT  = 000020
NUMRIG  003231R     025 RANERR= 104411          R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
NUMWID  003226R     025 RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
NUMZFL  003235R     025 RARGCV  003220R     025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
NYIERR= 104432          RARGFL  003202R     025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
N.BLSZ= 000036          RARGLE  003216R     025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DESC= 000020          RARGLN  003210R     025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DEV  = 000032         RARGPT  003206R     025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DEVN= 000034          RARGSF  003214R     025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.DID  = 000024         RARGTY  003212R     025 SIDAMN= 104433           S.AB   = 177776          T.VAL  = 100000
N.DVNM= 000032          RARGVA  003222R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAR  = 001000
N.EXT  = 000014         RBASE   005074R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER  = 177400
N.FID  = 000000         RCOUNR  005112R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.FNAM= 000006          RCOUNT  005110R     025 SITP    007144R     025 S.CHR1= 000003          VALERR= 104413
N.FTYP= 000014          REDCNT  005070R     025 SIWARN= 000200           S.FATT= 000016          VARBLE  007400R     025
N.FVER= 000016          REDIGN  005066R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLS  000000R     025
N.NAME= 000006          REDOFF  005072R     025 SI.BIT= 177774           S.FLB  = 040000          VERNUM  000000R     025
N.NEXT= 000022          REENTR  003060R     025 SI.LEN= 177772           S.FNAM= 000006          VRSION= 000002
N.PPN  = 000024         RESBIT  003142R     025 SI.RET= 177772           S.FNB  = 000036          WIDTH   000072R     025
N.STAT= 000020          RESCVT  003152R     025 SI.VAL= 177776           S.FNBW= 000017          WROUND  000020R     025
N.UNIT= 000034          RESERR= 104440          SKPSIZ  005134R     025 S.FNTY= 000004          WSDATE  000136R     025
OBUFLN= 001000          RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSFULL= 104403
OCHAN   005052R     025 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSNAME  000074R     025
OMODE   000402R     023 RESSFT  003146R     025 STKINC= 000774           S.LINK= 177774          WSTIME  000132R     025
OPTMP0  005066R     025 RESTYP  003144R     025 STOP   = 104420          S.NFEN= 000020          $DSW   = ****** GX
OPTMP1  005070R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP2  005072R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP3  005074R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP5  005100R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP6  005102R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSX   = ****** G
OPTMP7  005104R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          $RT11  = ****** G
OPTMP8  005106R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .BOOLE= 000002
OPTMP9  005110R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .CHARA= 000000
OPTM10  005112R     025 RSXERR  000062RG    020 SYMTP   006272R     025 TURKEY= 104436          .DEF.  = 000002
OPTM14  005026R     025 RTOP    005070R     025 SYMWRN= 000054           TYPE   = 000166          .FLOAT= 000006
ORIGIN  000030R     025 RT11   = 000000          SYNERR= 104407          T.BOL  = 020000          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003          SYSERR= 104402          T.CON  = 004000          .INTEG= 000004
OVROPR= 000000          R.FIX  = 000001          SYSTAT  000000R     024 T.DIM  = 000100          .MOLUN  000404R     023
PATCH   007344R     025 R.LCT  = 000002          SY.BAD= 000001          T.FUN  = 010000          ...GBL= 000000
PRECSN  000052R     025 R.LN   = 000002          SY.BRK= 100000          T.LIST= 000400          ...TPC= 000020
RANDAC  003064R     025 R.LNUM= 000010

. ABS.  000000      000
        000000      001
TABLES  00000C      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

```
BRKCHK  MACRO M1113  08-FEB-80 10:54  PAGE 47-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000132      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12954 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:48
BRKCHK,DM1:[305,301]BRKCHK/-SP=APLRX1,APLDEF,BRKCHK,END
```

FILEID**CARVE



.MAIN. MACRO M1113 08-FEB-80 10:55
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    CARVE            RETURN UNUSED DATA ON END OF ARRAY
```

CARVE MACRO M1113 08-FEB-80 10:55 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M
                
                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP
                
                000004          FPPLEN  =  4.           ;SINGLE PRECISION
                
                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CARVE   MACRO M1113  08-FEB-80 10:55  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CARVE
                                .IDENT  \V02.03\

                                .SBTTL  CARVE           RETURN UNUSED DATA ON END OF ARRAY

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'CARVE' CHOPS OFF AND RETURNS TO FREE CORE, ANY EXTRA UNUSED
                                ;       GARBAGE ON THE END OF AN ARRAY.
                                ;
                                ; IMPLICIT INPUTS:
                                ;       ON TOP OF SI -  ARRAY TO BE 'CARVED'
                                ;--
000000                                  .PSECT  CODE

000000                          CARVE::
000000  016501  177776                  MOV     SI.VAL(SI),R1   ;Point to Array Block
000004  005761  000002                  TST     A.DIMS(R1)      ;Is it an Array?
000010  002072                          BGE     5$              ;No, SYSTEM ERROR
000012  005761  000004                  TST     A.ALEN(R1)      ;Anything There?
000016  001411                          BEQ     1$              ;Nope, is Null Array
000020  020111                          CMP     R1,(R1)         ;Is the Data in the Block?
000022  101007                          BHI     1$              ;Nope, Not Here
000024  010100                          MOV     R1,R0           ;Copy Pointer
000026  014002                          MOV     -(R0),R2        ;Get Count of Words in Block
000030  005202                          INC     R2              ;Decrement (But it's Negative)
000032  160200                          SUB     R2,R0           ;Point to End of Block
000034  160200                          SUB     R2,R0           ;  (Offset in Words)
000036  020011                          CMP     R0,(R1)         ;Is Data in Block
000040  101011                          BHI     2$              ;Yes, Oh Frabjus Day
                                ;
                                ; Handle Header without Data
                                ;
000042  116100  000002          1$:     MOVB    A.DIMS(R1),R0   ;Get # of Dimensions
000046                                  CONINC  R0              ;Cheat for Scalar
000052  006300                          ASL     R0              ;Times 2 (Byte Count)
000054  060100                          ADD     R1,R0           ;Include Base Pointer
000056  062700  000006                  ADD     #A.DIM1,R0      ;And Size of Block
000062  000420                          BR      3$              ;...Go
                                ;
                                ; Handle Block With Data
                                ;
000064  016100  000004          2$:     MOV     A.ALEN(R1),R0   ;Get the Total Length
000070  116102  000003                  MOVB    A.BITS(R1),R2   ;Get the Source Type
000074  042702  177771                  BIC     #^C<A.NUM>,R2   ;Clear Garbage Bits
000100                                  ASH     SIZTAB(R2),R0   ;Compute Total Size
000114  005200                          INC     R0              ;Perform "Ceiling"
000116  042700  000001                  BIC     #1,R0           ;  for Word Pointer
000122  061100                          ADD     (R1),R0         ;Add Pointer
                                ;
                                ; Now to Extract the Data
                                ;
000124  014102                  3$:     MOV     -(R1),R2        ;Get (Negative) Block Size
000126  005402                          NEG     R2              ;Make Offset Positive
000130  010003                          MOV     R0,R3           ;Get Pointer to the End
000132  160103                          SUB     R1,R3           ;Compute Needed Length
000134  006203                          ASR     R3              ;  In Words
```

CARVE RETURN UNUSED DATA ON END OF ARRAY

```
     58 000136  005203                          INC     R3              ;Add 1 Extra (Cheat)
     59 000140  010304                          MOV     R3,R4           ;Copy the Length
     60 000142  005404                          NEG     R4              ;Negate For allocated Size
     61 000144  160203                          SUB     R2,R3           ;Get (Negative) Extra Length
     62 000146  020327  177772                  CMP     R3,#-6.         ;Less Than 6 Words?
     63 000152  003010                          BGT     4$              ;Yes, Don't Do Anything
     64 000154  006302                          ASL     R2              ;Size of Original (In Bytes)
     65 000156  060102                          ADD     R1,R2           ;Point Past Next Block
     66 000160  010411                          MOV     R4,(R1)         ;Set New Length
     67 000162  010342                          MOV     R3,-(R2)        ;Set Length of Extra
     68 000164  010420                          MOV     R4,(R0)+        ;Set Top of Data Block
     69 000166  010320                          MOV     R3,(R0)+        ;Set Bottom of Garbage Block
     70 000170                                  CALL    RETCOR          ;Return Extra Memory
     71 000174                          4$:     RETURN                  ;Finished
     72
     73 000176  104402                  5$:     SYSERR                  ;Wasn't an Array
```

```
CARVE   MACRO M1113  08-FEB-80 10:55  PAGE 46
CARVE           RETURN UNUSED DATA ON END OF ARRAY

      1         000001                          .END
```

```
CARVE   MACRO M1113  08-FEB-80 10:55  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.BLOK= 000064  D.DECD= 000002  EIS    = 000000
AC1    =%000001          CHRLBR= 000073  C.BLSZ= 000140  D.DIV = 000000  ELESIZ  005132R     025
AC2    =%000002          CHRLPR= 000072  C.BUFF= 177760  D.DMD = 000010  ENDCOR  000004R     025
AC3    =%000003          CHRMNS= 000137  C.BUFL= 177762  D.DQAD= 177777  EOF    = 104475
AC4    =%000004          CHRMUL= 000075  C.CCNT= 177766  D.DROP= 000002  ERRMAX= 000166
AC5    =%000005          CHRNEG= 000100  C.CPTR= 177764  D.ENCD= 000002  ERRNUM  004772R     025
ALLINT  003274R     025  CHRPDL= 000271  C.CZ  = 001000  D.EQ  = 000000  ERRPTR  004770R     025
ARYRET  003276R     025  CHRPLU= 000055  C.DIG = 000040  D.EXPD= 000002  ERR0    004774R     025
ASCQUO= 000042           CHRPWR= 000120  C.FASW= 020000  D.FGET= 000005  EXEBUF  005054R     025
A.ALEN= 000004           CHRQST= 000121  C.FLEN= 000010  D.FMT = 000004  E$CT    005056R     025
A.ARG = 000040           CHRQUO= 000113  C.FLSH= 004000  D.FPNT= 000005  E$PT    005060R     025
A.ASB = 000200           CHRRBR= 000047  C.FRC = 000001  D.FPUT= 000005  FA.APD= 000100
A.BITS= 000003           CHRRPR= 000042  C.IDEN= 000010  D.GE  = 000000  FA.CRE= 000010
A.CON = 000020           CHRSEM= 000074  C.NAME= 000102  D.GT  = 000000  FA.DLK= 001000
A.DIMS= 000002           CHRSMO= 000112  C.NOBF= 000400  D.IBM = 000002  FA.ENB= 100000
A.DIM1= 000006           CHRSPA= 000040  C.NUM = 000020  D.IND = 000002  FA.EXC= 002000
A.NUM = 000006           CHRUPA= 000131  C.OP  = 000004  D.INNR= 000013  FA.EXT= 000004
A.PTR = 000000           CHR.BS= 000010  C.OPN = 010000  D.INTR= 000006  FA.NSP= 000100
A.TMP = 000010           CHR.CC= 000003  C.RDLN= 177776  D.LE  = 000000  FA.NSR= 000200
BADDIR= 104463           CHR.CR= 000015  C.REWT= 002000  D.LOG = 000001  FA.POS= 010000
BADERR= 104437           CHR.CZ= 000032  C.RND = 000010  D.LT  = 000000  FA.RD  = 000001
BLZBUB  005116R     025  CHR.GT= 000046  C.SBIT= 177756  D.MAX = 000000  FA.RWD= 004000
BRSELF  005046R     025  CHR.LF= 000012  C.SPFX= 000100  D.MEM = 000002  FA.SEQ= 040000
CARVE   000000RG    020  CHR.LT= 000043  C.SPGL= 000200  D.MIN = 000000  FA.SHR= 000040
CCNT    005072R     025  CHR.SI= 000017  C.SPLX= 000001  D.MUL = 000000  FA.TMP= 000020
CHANEX  003060R     025  CHR.SL= 000057  C.SPRS= 000002  D.NE  = 000000  FA.WCK= 020000
CHAN00  000256R     025  CHR.SO= 000016  C.SRC = 177774  D.NN  = 000000  FA.WRT= 000002
CHAN01  000330R     025  CHR.TB= 000011  C.SRCP= 000024  D.NR  = 000000  FD.BLK= 000010
CHAN02  000512R     025  CHR.UL= 000106  C.TT  = 000004  D.OR  = 000000  FD.CCL= 000002
CHAN03  000674R     025  CHR..A= 000141  C.WIDC= 177770  D.PLUS= 000000  FD.COM= 020000
CHAN04  001056R     025  CHR..E= 000145  C.WIDR= 177772  D.PSUB= 000006  FD.CR = 000002
CHAN05  001240R     025  CHR..S= 000163  DEBUG = 000000  D.PWR = 000001  FD.DIR= 000010
CHAN06  001422R     025  CHR..T= 000164  DEFERR= 104405  D.QQAD= 000000  FD.FTN= 000001
CHAN07  001604R     025  CHR..U= 000165  DEVIL   005114R     025 D.QUAD= 000000  FD.FTI= 040000
CHAN08  001766R     025  CHR..Z= 000172  DEVNTF= 104470  D.RES = 000000  FD.INS= 000010
CHAN09  002150R     025  CHR..0= 000060  DIGITS  000006R     025 D.RHO = 000000  FD.ISP= 002000
CHAN10  002332R     025  CHR..1= 000061  DIMAX = 000020  D.ROT = 000003  FD.MNT= 100000
CHAN11  002514R     025  CH.AND= 000001  DIMBLK  003112R     025 D.SCON= 000006  FD.OSP= 004000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006  FD.PLC= 000004
CHAN13  000022R     023  CMDMAX= 000010  DIMTYP  003116R     025 D.SUB = 000000  FD.PRN= 000004
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466  D.SUBS= 000006  FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417  D.TAKE= 000002  FD.RAH= 000001
CHRAND= 000051           COMCNT  005126R     025 DPTERR= 104416  D.TRAN= 000004  FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425      DYDCNT  003224R     025 D.UNON= 000006  FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R     025 D.AND = 000000  D.$MAX= 000000  FD.RWM= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.BXCT= 000007  D.$MIN= 000000  FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R     025 D.BXIO= 000007  D.$MUL= 000000  FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R     025 D.BXNL= 000007  D.$PLU= 000000  FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200          D.BXPP= 000007  D.$RES= 000000  FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010          D.BXPW= 000007  D.$SUB= 000000  FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020          D.BXRL= 000007  D.1COM= 000001  FF.NV = 000003
CHRDLR= 000176           CS.IDN= 000040          D.CIRC= 000001  D.1EXP= 000002  FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100          D.COMB= 000001  D.1ROT= 000003  FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R     025 D.COMP= 000001  EDICHR  005063R     025 FF.RWF= 000006
CHREP = 000105           CTMAX   000212R     025 D.CONT= 000006  EDICNT  005064R     025 FF.SPC= 000004
CHRERR= 104431           CTRMAX  000172R     025 D.CTLM= 000000  EDIFLG  005062R     025 FHEADR  005012R     025
CHRFIL= 000177           C.ALOC= 000162          D.DEAL= 000003  EDIT   = 000003         FHRET   005040R     025
```

SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FIS   = 000000          F.FNDF= 000010          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FIXPTR  003074R 025     F.FUN = 000010          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FLAGS   004764R 025     F.FVER= 000120          INDERR= 104410          M.ABS = 000000      M.SUBR= 000001
FLCBUF  003250R 025     F.HEAD= 000006          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLINE   005014R 025     F.HIBK= 000004          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLOEXP  003270R 025     F.LBL = 000020          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLONEG  003272R 025     F.LINE= 000012          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLSWTB  000222R 025     F.LOC = 000004          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FNINSI  005044R 025     F.LUN = 000042          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.RD = 000001          F.MON = 000002          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FPMODE  000042R 025     F.NREC= 000030          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPP   = 000000          F.OPN = 000010          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FREBLK  007400R 025     F.PREV= 000000          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FUZZ    000142R 025     F.PRNT= 100000          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FXPTR1  003076R 025     F.PROT= 010000          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVI = 000200          F.RATT= 000001          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F.ACTL= 000076          F.READ= 004000          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BDB = 000070          F.RET = 177772          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R 025 M.GDWN= 000003      ND.$ES= 000020
F.BKDN= 000026          F.R2  = 000014          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.BKDS= 000020          F.R3  = 000016          LFTOP   005066R 025 M.IBM = 000003      ND.$NS= 000040
F.BKEF= 000050          F.R4  = 000020          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKST= 000024          F.SPDV= 000072          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BR  = 000020          F.STBK= 000036          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.CHR = 000075          F.STK = 000022          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CODE= 000010          F.STP = 000001          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DSPT= 000044          F.TRC = 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.EFBK= 000010          F.URBD= 000020          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFN = 000050          F.VAL = 177776          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.END = 002000          F.VBN = 000064          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.ERR = 000052          GNOME   003102R 025     L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
F.FACC= 000043          HASHC   005136R 025     MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FFBY= 000014          HASHTB  005140R 025     MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
F.FLAG= 000002          HSHMSK= 177601          MONBLK  003120R 025 M.RDCT= 000011      NUMQFL  003234R 025
F.FLB = 000100          HSHSIZ= 000200          MONCVT  003132R 025 M.REVS= 000003      NUMRIG  003231R 025
F.FNAM= 000110          IASH  = ****** GX       MONLEN  003130R 025 M.ROLL= 000000      NUMWID  003226R 025
```

```
CARVE   MACRO M1113  08-FEB-80 10:55  PAGE 46-3
SYMBOL TABLE

NUMZFL  003235R     025 RARGBL  003204R     025 R.NEXT= 000006          SY.BS  = 000010          T.LIT  = 000020
NYIERR= 104432          RARGCV  003220R     025 R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
N.BLSZ= 000036          RARGFL  003202R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
N.DESC= 000020          RARGLE  003216R     025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
N.DEV  = 000032         RARGLN  003210R     025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DEVN= 000034          RARGPT  003206R     025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DID  = 000024         RARGSF  003214R     025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DVNM= 000032          RARGTY  003212R     025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.EXT  = 000014         RARGVA  003222R     025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.FID  = 000000         RBASE   005074R     025 SIDAMN= 104433           S.AB   = 177776          T.VAL  = 100000
N.FNAM= 000006          RCOUNR  005112R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAR  = 001000
N.FTYP= 000014          RCOUNT  005110R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER  = 177400
N.FVER= 000016          REDCNT  005070R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.NAME= 000006          REDIGN  005066R     025 SITP    007144R     025 S.CHR1= 000003          VALERR= 104413
N.NEXT= 000022          REDOFF  005072R     025 SIWARN= 000200           S.FATT= 000016          VARBLE  007400R     025
N.PPN  = 000024         REENTR  003060R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLS  000000R     025
N.STAT= 000020          REGTYP= 000000          SI.BIT= 177774           S.FLB  = 040000          VERNUM  000000R     025
N.UNIT= 000034          RESBIT  003142R     025 SI.LEN= 177772           S.FNAM= 000006          VRSION= 000002
OBUFLN= 001000          RESCVT  003152R     025 SI.RET= 177772           S.FNB  = 000036          WIDTH   000072R     025
OCHAN   005052R     025 RESERR= 104440          SI.VAL= 177776           S.FNBW= 000017          WROUND  000020R     025
OMODE   000402R     023 RESLEN  003150R     025 SKPSIZ  005134R     025 S.FNTY= 000004          WSDATE  000136R     025
OPTMP0  005066R     025 RESPTR  003140R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSFULL= 104403
OPTMP1  005070R     025 RESSFT  003146R     025 SPINIT  000002R     024 S.LEN  = 000002          WSNAME  000074R     025
OPTMP2  005072R     025 RESTYP  003144R     025 STKINC= 000774           S.LINK= 177774          WSTIME  000132R     025
OPTMP3  005074R     025 RESULT  003136R     025 STOP   = 104420          S.NFEN= 000020          $FIS   = ****** G
OPTMP4  005076R     025 RESVAL  003154R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FPP   = ****** G
OPTMP5  005100R     025 RETCOR= ****** GX       SUBDCT  003106R     025 S.VAL  = 000000          $FPPLN= ****** G
OPTMP6  005102R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP7  005104R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP8  005106R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP9  005110R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RT11  = ****** G
OPTM10  005112R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          .BOOLE= 000002
OPTM14  005026R     025 RSTS   = 000000          SYMTBF= 104404           TTYBUF  004140R     025 .CHARA= 000000
ORIGIN  000030R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .DEF.  = 000002
OUTPTR  000066R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436          .FLOAT= 000006
OVROPR= 000000          RT11   = 000000          SYMWRN= 000054           TYPE   = 000166          .FSRCB= ****** G
PATCH   007344R     025 R.BITS= 000003          SYNERR= 104407           T.BOL  = 020000          .INTEG= 000004
PRECSN  000052R     025 R.FIX  = 000001          SYSERR= 104402           T.CON  = 004000          .MOLUN  000404R     023
RANDAC  003064R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100          ...GBL= 000000
RANDY   003062R     025 R.LN   = 000002          SY.BAD= 000001           T.FUN  = 010000          ...TPC= 000020
RNVERR= 104411          R.LNUM= 000010          SY.BRK= 100000           T.LIT  = 000400

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
CARVE   MACRO M1113  08-FEB-80 10:55  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000200      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12934 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:41
CARVE,DM1:[305,301]CARVE/-SP=APLRX1,APLDEF,CARVE,END
```

```
**FILE**ID**CHRTAB
```

```
  CCCCCCCC    HH      HH    RRRRRRRR      TTTTTTTTTT      AAAAAA      BBBBBBBB
  CCCCCCCC    HH      HH    RRRRRRRR      TTTTTTTTTT      AAAAAA      BBBBBBBB
CC            HH      HH    RR      RR        TT        AA      AA    BB      BB
CC            HH      HH    RR      RR        TT        AA      AA    BB      BB
CC            HH      HH    RR      RR        TT        AA      AA    BB      BB
CC            HH      HH    RR      RR        TT        AA      AA    BB      BB
CC            HHHHHHHHHH    RRRRRRRR          TT        AA      AA    BBBBBBBB
CC            HHHHHHHHHH    RRRRRRRR          TT        AA      AA    BBBBBBBB
CC            HH      HH    RR   RR           TT        AAAAAAAAAA    BB      BB
CC            HH      HH    RR   RR           TT        AAAAAAAAAA    BB      BB
CC            HH      HH    RR     RR         TT        AA      AA    BB      BB
CC            HH      HH    RR     RR         TT        AA      AA    BB      BB
  CCCCCCCC    HH      HH    RR       RR       TT        AA      AA    BBBBBBBB
  CCCCCCCC    HH      HH    RR       RR       TT        AA      AA    BBBBBBBB
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 10:57
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     5    DEFINITIONS     MACRO USED TO GENERATE TABLES
```

CHRTAB MACRO M1113 08-FEB-80 10:57
TABLE OF CONTENTS

45- 68 DEFINITIONS TABLE ORIGINS
46- 133 TABLES CHARACTER TABLES

CHRTAB MACRO M1113 08-FEB-80 10:57 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  CHRTAB
                                .IDENT  \V02.03\


                                .SBTTL  DEFINITIONS     MACRO USED TO GENERATE TABLES

                                .MACRO  CHR BITS,SPEC,TOASCI,MNASCI,OVAPL1,OVAPL2
                                        .NLIST
                                ;---------------------------------- CHRTAB ----------------------------------
                                        .PSECT CHRTBL
                                        .BYTE BITS+0
                                ;---------------------------------- VTAB ------------------------------------
                                        .PSECT VTBL
                                        .IF B <SPEC>
                                        .BYTE -1
                                        .IFF
                                        .BYTE SPEC
                                        .ENDC
                                ;---------------------------------- ASCTAB ----------------------------------
                                        .PSECT ASCTBL
                                        .IF B <TOASCI>
                                        .WORD TMPTAG
                                        .IFF
                                        .WORD TOASCI+LC.KLD
                                        .ENDC
                                        .NCHR ASCII,<TOASCI>
                                ;---------------------------------- MNTAB & MNTRTB --------------------------
                                        .IF NB <MNASCI>
                                        .PSECT MNTBL
                                        .WORD 'MNASCI
                                        .PSECT MNTRTL
                                        .BYTE TMPTAG
                                        .IFF
                                        .IF EQ ASCII-3
                                        .PSECT MNTBL
                                        .WORD TOASCI
                                        .PSECT MNTRTL
                                        .BYTE TMPTAG
                                        .ENDC
                                        .ENDC
                                ;---------------------------------- TRNTAB ----------------------------------
                                        .IF EQ ASCII-2
                                        .PSECT TRNTBL
                                        TRNADR TOASCI+LC.KLD,TMPTAG
                                        .IFF
                                        .IF LE TMPTAG-40
                                        .PSECT TRNTBL
                                        TRNADR TMPTAG,TMPTAG
                                        .ENDC
                                        .ENDC
                                ;---------------------------------- OVRTAB ----------------------------------
                                        .IF NB <OVAPL1>
                                        .PSECT OVRTBL
                                        .WORD OVAPL1*256.+OVAPL2
                                        .ENDC

                                        TMPTAG=TMPTAG+1
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 44-1
DEFINITIONS     MACRO USED TO GENERATE TABLES

                                        .LIST
                                .ENDM

                                .MACRO TRNADR WHERE,WHAT
                                        .=.+<WHERE-CURPTR>
                                        CURPTR=WHERE+1
                                        .BYTE WHAT
                                .ENDM
```

```
                                .SBTTL  DEFINITIONS     TABLE ORIGINS


000000                                  .PSECT CHRTBL
000000                          CHRTAB::
                                ; THE OFFSET IN THIS TABLE OF AN APL CHARACTER CONTAINS
                                ; BITS DESCRIBING THAT CHARACTER.


000000                                  .PSECT VTBL
000000                          VTAB::
                                ;THIS TABLE CONTAINS TABLE OFFSETS (SPECIALTY NUMBERS) FOR
                                ;APL CHARACTERS.  FOR EXAMPLE, THE OFFSET IN THIS TABLE OF
                                ;AN APL OPERATOR CONTAINS THE OPERATOR NUMBER FOR THAT OP.
                                ;THE OFFSET IN THIS TABLE OF A SPECIAL CHARACTER (SUCH
                                ;AS A SPECIAL LEX CHARACTER) CONTAINS THE OFFSET IN THE
                                ;TABLE WHICH POINTS TO THE ROUTINE TO HANDLE THAT CHARACTER.


000000                                  .PSECT MNTBL
000000                          MNTAB::
                                ;THIS TABLE CONTAINS ASCII MNEMONICS FOR SOME APL
                                ;CHARACTERS.  THIS TABLE IS SEARCHED VIA 'BINSCH'
                                ;
                                ;*** NOTE *** THE OFFSET OF THE MNEMONIC IN THIS
                                ;TABLE IS USED TO LOOK UP THE CORRESPONDING APL
                                ;CHARACTER IN 'MNTRTB', SO ANY CHANGES IN THIS
                                ;TABLE MUST ALSO BE MADE IN 'MNTRT'.


000000                                  .PSECT MNTRTL
000000                          MNTRTB::
                                ;'MNTRTB' CONTAINS THE APL CHARACTER EQUIVALENTS OF
                                ;THE ASCII MNEMONICS IN 'MNTAB'.  A MNEMONIC IS FIRST
                                ;LOOKED UP IN 'MNTAB', THE OFFSET IS DIVIDED BY 2, AND
                                ;THE CORRESPONDING CHAR IS THEN FOUND IN THIS TABLE.
                                ;NOTE THAT THIS IS A BYTE TABLE, WHILE 'MNTAB' IS A WORD
                                ;TABLE!


000000                                  .PSECT ASCTBL
000000                          ASCTAB::
                                ;THE OFFSET (IN WORDS!!!!) IN THIS TABLE OF AN APL
                                ;CHARACTER CONTAINS EITHER THE SINGLE ASCII CHAR
                                ;REPRESENTING THAT APL CHAR, OR THE 2 CHARS WHICH
                                ;ARE THE MNEMONIC FOR THAT APL CHAR.  'OUTCHR' USES
                                ;THIS TABLE TO CONVERT THE CHARS IF 'TMODE' IS SET.


000000                                  .PSECT TRNTBL
000000                          TRNTAB::
           000000               CURPTR=0
                                ;THE INDEX IN THIS TABLE OF A SINGLE ASCII CHARACTER
                                ;CONTAINS THE CORRESPONDING APL CHARACTER FOR THAT
                                ;CHARACTER.  IN GENERAL, THE TRANSLATIONS ARE OBVIOUS,
                                ;BUT IN SOME I WAS MORE CREATIVE.  NOTE THAT '%' IS
                                ;DIVIDE, AND THAT '#' IS MULTIPLY.
```

```
    125
    126
    127 000000                          .PSECT OVRTBL
    128 000000                      OVRTAB::
    129                             ;THIS TABLE CONTAINS ALL LEGAL APL OVERSTRUCK CHARACTERS.
    130                             ;THE CHARACTERS ARE ORDERED WITH THE LOWEST VALUED
    131                             ;CHARACTER IN THE UPPER BYTE.
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 46
TABLES          CHARACTER TABLES


                                .SBTTL  TABLES          CHARACTER TABLES

                                ;THE FOLLOWING CODE IS USED TO GENERATE VARIOUS CHARACTER TABLES.
                                ;IT IS SET UP SO THAT THE OVERSTRUCK APL CHARACTERS MUST BE AT THE
                                ;END OF THE LIST.  UTMOST CARE SHOULD BE TAKEN WHEN CHANGING THIS
                                ;SECTION AS THE DEFINITIONS OF SYMBOLS CAN GET SCREWED ROYAL LIKE I DID A
                                ;FEW TIMES.  BELIEVE ME, IT'S A REAL PAIN!

                                ;THE ARGUMENTS FOR THE MACRO ARE: (THE VALUE ASSUMED IS IN PARANTHESIS)
                                ;
                                ;       (0)     (-1)    (TMPTAG) --     --,--
                                ;MACRO  CHRTAB  VTAB    ASCTAB  MNTAB&  OVERSTRUCK
                                ;NAME   BITS    VALUE   EQUIVS  MNTRTB  CHR VALUES
                                ;

          000000                LC.KLD=0        ;SET TO 40 FOR 'ASCTAB' ENTRIES IN LOWER CASE
          000000                TMPTAG=0        ;CHARACTERS START AT ZERO

000000                          CHR                                             ;NULL
000001                          CHR                                             ;CTRL A
000002                          CHR                                             ;CTRL B
000003                          CHR     C.SPGL, 10                              ;CTRL C
000004                          CHR     C.SPGL, 12                              ;CTRL D
000005                          CHR                                             ;CTRL E
000006                          CHR                                             ;CTRL F
000007                          CHR                                             ;BELL
000010                          CHR     C.SPFX, 4                               ;BACKSPACE
000011                          CHR                                             ;TAB
000012                          CHR     C.SPFX+C.SPGL, 6                        ;LINE FEED
000013                          CHR                                             ;CTRL K
000014                          CHR                                             ;FORM FEED
000015                          CHR     C.SPGL+C.SPLX+C.SPRS+C.SPFX, 0          ;CARRIGE RETURN
000016                          CHR                                             ;CTRL N
000017                          CHR                                             ;CTRL O
000020                          CHR                                             ;CTRL P
000021                          CHR                                             ;CTRL Q
000022                          CHR                                             ;CTRL R
000023                          CHR                                             ;CTRL S
000024                          CHR                                             ;CTRL T
000025                          CHR     C.SPGL, 2                               ;CTRL U
000026                          CHR                                             ;CTRL V
000027                          CHR                                             ;CTRL W
000030                          CHR                                             ;CTRL X
000031                          CHR                                             ;CTRL Y
000032                          CHR     C.SPGL, 14                              ;CTRL Z
000033                          CHR     C.SPGL, 16                              ;ESCAPE (ALTMODE)
000034                          CHR                                             ;CTRL \
000035                          CHR                                             ;CTRL ]
000036                          CHR                                             ;CTRL ^
000037                          CHR                                             ;CTRL _
```

```
                                CHR     C.SPFX+C.SPRS, 2                        ;SPACE
                                CHR     ,       ,       'DD                     ;DIERESIS
                                CHR     ,       ,       ')                      ;)
                                CHR     C.OP,   24,     74,     LT              ;<
                                CHR     C.OP,   26,     'LE                     ;<=
        000030                  OP.EQU==30
                                CHR     C.OP,   30,     '=                      ;=
                                CHR     C.OP,   32,     76,     GT              ;>
                                CHR     ,       ,       ']                      ;]
                                CHR     C.OP,   40,     'OR                     ;OR
                                CHR     C.OP,   42,     '&,     AN              ;AND
        000034                  OP.NEQ==34
                                CHR     C.OP,   34,     'NE                     ;NOT EQUAL
                                CHR     C.OP,   10,     '%,     DV              ;DIVIDE
                                CHR     C.OP,   56,     54                      ;COMMA
        000002                  OP.PLU==2
                                CHR     C.OP,   2,      '+                      ;+
                                CHR     C.SPLX, 6,      '.                      ;.
        000060                  OP.RED==60
                                CHR     C.OP,   60,     '/,                     ;/
        000061                  C.TMP=C.NUM+C.DIG+C.SPLX
                                CHR     C.TMP,  12,     '0                      ;0
                                CHR     C.TMP,  12,     '1                      ;1
                                CHR     C.TMP,  12,     '2                      ;2
                                CHR     C.TMP,  12,     '3                      ;3
                                CHR     C.TMP,  12,     '4                      ;4
                                CHR     C.TMP,  12,     '5                      ;5
                                CHR     C.TMP,  12,     '6                      ;6
                                CHR     C.TMP,  12,     '7                      ;7
                                CHR     C.TMP,  12,     '8                      ;8
                                CHR     C.TMP,  12,     '9                      ;9
                                CHR     ,       ,       '(                      ;(
                                CHR     ,       ,       '[                      ;[
                                CHR     ,       ,       73                      ;
        000006                  OP.MUL==6
                                CHR     C.OP,   6,      '#                      ;TIMES
                                CHR     ,       ,       ':                      ;:
        000062                  OP.SCN==62
                                CHR     C.OP,   62,     '\                      ;\
                                CHR     C.NUM+C.SPLX, 12,       'NG             ;NEGATE
                                CHR     ,       ,       'A_                     ;ALPHA
                                CHR     C.OP,   64,     'DE                     ;DECODE
                                CHR     C.OP,   66,     'DU                     ;DOWN U
        000014                  OP.FLR==14
                                CHR     C.OP,   14,     'FL                     ;FLOOR
                                CHR     C.OP,   70,     'EP                     ;EPSILON
                                CHR     C.IDEN, .       'US                     ;UNDERSCORE
                                CHR     ,       ,       'DL                     ;DEL
                                CHR     C.IDEN+C.SPLX, 10,      'LD             ;DELTA
                                CHR     C.OP,   72,     'IO                     ;IOTA
                                CHR     ,       44,     'SO                     ;SMALL O
                                CHR     C.SPLX, 2,      ''                      ;QUOTE
                                CHR     C.OP,   74,     'BX                     ;QUAD
        000012                  OP.RES==12
                                CHR     C.OP,   12,     'AB                     ;RESIDUE
        000076                  OP.ENC==76
                                CHR     C.OP,   76,     'EN                     ;ENCODE
```

TABLES CHARACTER TABLES

```
241 000026                      CHR     C.OP,   52.,    'LO             ;LARGE O
242 000027                      CHR     C.OP,   20,     '*              ;EXPONENT
243 000053                      CHR     C.OP,   100,    '?              ;RANDOM
244             000102          OP.RHO==102
245 000100                      CHR     C.OP,   102,    'RO             ;RHO
246             000016          OP.CEI==16
247 000030                      CHR     C.OP,   16,     'CE             ;CEILING
248 000031                      CHR     C.OP,   50,     'NT             ;NOT
249 000032                      CHR     C.OP,   104,    'DA             ;DROP
250 000033                      CHR     C.OP,   106,    'UU             ;UP U
251 000034                      CHR     C.OP,   110,    'OM             ;OMEGA
252 000035                      CHR     C.OP,   234,    'LU             ;LEFT U (CONTAINS)
253 000036                      CHR     C.OP,   114,    '^,     TK      ;TAKE
254 000137                      CHR     C.OP,   236,    'RU             ;RIGHT U (SUBSET)
255 000040                      CHR     ,       0,      '               ;ASSIGN
256 000140                      CHR     C.OP,   120,    'RK             ;RIGHT TACK
257 000041                      CHR     ,       0,      'GO             ;BRANCH
258 000042                      CHR     C.OP,   36,     'GE             ;>=
259 000043                      CHR     C.OP,   4,      '-              ;SUBTRACT
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 48
TABLES          CHARACTER TABLES

    261 000056                          CHR             116.    'DM             ;DIAMOND
    262             000011              C.$=C.IDEN+C.SPLX
    263 000044                          CHR     C.$.    10.     'A              ;A
    264 000102                          CHR     C.$.    10.     'B              ;B
    265 000103                          CHR     C.$.    10.     'C              ;C
    266 000104                          CHR     C.$.    10.     'D              ;D
    267 000105                          CHR     C.$.    10.     'E              ;E
    268 000106                          CHR     C.$.    10.     'F              ;F
    269 000107                          CHR     C.$.    10.     'G              ;G
    270 000110                          CHR     C.$.    10.     'H              ;H
    271 000111                          CHR     C.$.    10.     'I              ;I
    272 000112                          CHR     C.$.    10.     'J              ;J
    273 000113                          CHR     C.$.    10.     'K              ;K
    274 000114                          CHR     C.$.    10.     'L              ;L
    275 000115                          CHR     C.$.    10.     'M              ;M
    276 000116                          CHR     C.$.    10.     'N              ;N
    277 000117                          CHR     C.$.    10.     'O              ;O
    278 000120                          CHR     C.$.    10.     'P              ;P
    279 000121                          CHR     C.$.    10.     'Q              ;Q
    280 000122                          CHR     C.$.    10.     'R              ;R
    281 000123                          CHR     C.$.    14.     'S              ;S
    282 000124                          CHR     C.$.    14.     'T              ;T
    283 000125                          CHR     C.$.    10.     'U              ;U
    284 000126                          CHR     C.$.    10.     'V              ;V
    285 000127                          CHR     C.$.    10.     'W              ;W
    286 000130                          CHR     C.$.    10.     'X              ;X
    287 000131                          CHR     C.$.    10.     'Y              ;Y
    288 000132                          CHR     C.$.    10.     'Z              ;Z
    289 000133                          CHR     .       .       'LB             ;LEFT BRACE
    290 000045                          CHR     .       120.    'LK             ;LEFT TACK
    291 000046                          CHR     .       .       'RB             ;RIGHT BRACE
    292 000047                          CHR     .       .       '$              ;DOLLAR SIGN
    293
    294
    295 000045                          CHR     C.SPRS+C.SPGL.  4.      000177  ;FILLER
    296 000400                          CHR     C.SPRS+C.SPGL.  6.      '??+00  ;BEGINNING OF LINE
    297 000402                          CHR     C.SPRS+C.SPGL.  10.     '??+00  ;SYNTAX ERROR
    298 000404                          CHR     C.SPRS+C.SPGL.  .       '??+00  ;CHARACTER ERROR
    299 000406                          CHR     C.SPRS+C.SPGL.  16.     '??+00  ;IDENTIFIER
    300 000410                          CHR     C.SPRS+C.SPGL.  14.     '??+00  ;NUMERIC CONSTANT
    301 000412                          CHR     C.SPRS+C.SPGL.  12.     '??+00  ;LITTERAL CONSTNT
    302 000414                          CHR     C.SPRS+C.SPGL.  20.     '??+00  ;CHARACTER SKIP
    303 000416                          CHR     .       .       '??+00          ;<SPARE>
    304 000420                          CHR     .       .       '??+00          ;<SPARE>
    305 000422                          CHR     .       .       '??+00          ;<SPARE>
    306 000424                          CHR     .       .       '??+00          ;<SPARE>
    307 000426                          CHR     .       .       '??+00          ;<SPARE>
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 49
TABLES          CHARACTER TABLES

    309             000214          OVSTRK==TMPTAG
    310 000430                      CHR     C.OP,   44,     'NR,    ,       50,124  ;NOR
    311 000002                      CHR     C.OP,   46,     'NN,    ,       51,124  ;NAND
    312 000004                      CHR     C.OP,   122,    'DQ,    ,       53,114  ;MATRIX INVERT
    313             000217          CHRFAC==TMPTAG
    314 000006                      CHR     C.OP,   22,     '!,     ,       56,113  ;FACTORIAL
    315 000010                      CHR     C.OP,   124,    'CS,    ,       57,137  ;F.C. SLASH
    316 000012                      CHR     C.OP,   126,    'TR,    ,       77,117  ;TRANSPOSE
    317 000014                      CHR     C.OP,   130,    'CB,    ,       77,137  ;F.C. BACKSLASH
    318 000016                      CHR     C.OP,   132,    'IB,    ,       102,116 ;I BEAM
    319             000224          CHRLMP==TMPTAG
    320 000020                      CHR     C.SPLX, 4,      ''',    ,       103,112 ;LAMP
    321 000022                      CHR     C.$,    10,     'Z@,    ,       106,110 ;UNDERLINE DELTA
    322             000226          CHRULA==TMPTAG                                  ;DEFINE 'UNDERLINE A'
    323 000024                      CHR     C.$,    10,     'ZA,    ,       106,141 ;UNDLIN A
    324 000026                      CHR     C.$,    10,     'ZB,    ,       106,142 ;       B
    325 000030                      CHR     C.$,    10,     'ZC,    ,       106,143 ;       C
    326 000032                      CHR     C.$,    10,     'ZD,    ,       106,144 ;       D
    327 000034                      CHR     C.$,    10,     'ZE,    ,       106,145 ;       E
    328 000036                      CHR     C.$,    10,     'ZF,    ,       106,146 ;       F
    329 000040                      CHR     C.$,    10,     'ZG,    ,       106,147 ;       G
    330 000042                      CHR     C.$,    10,     'ZH,    ,       106,150 ;       H
    331 000044                      CHR     C.$,    10,     'ZI,    ,       106,151 ;       I
    332 000046                      CHR     C.$,    10,     'ZJ,    ,       106,152 ;       J
    333 000050                      CHR     C.$,    10,     'ZK,    ,       106,153 ;       K
    334 000052                      CHR     C.$,    10,     'ZL,    ,       106,154 ;       L
    335 000054                      CHR     C.$,    10,     'ZM,    ,       106,155 ;       M
    336 000056                      CHR     C.$,    10,     'ZN,    ,       106,156 ;       N
    337 000060                      CHR     C.$,    10,     'ZO,    ,       106,157 ;       O
    338 000062                      CHR     C.$,    10,     'ZP,    ,       106,160 ;       P
    339 000064                      CHR     C.$,    10,     'ZQ,    ,       106,161 ;       Q
    340 000066                      CHR     C.$,    10,     'ZR,    ,       106,162 ;       R
    341 000070                      CHR     C.$,    10,     'ZS,    ,       106,163 ;       S
    342 000072                      CHR     C.$,    10,     'ZT,    ,       106,164 ;       T
    343 000074                      CHR     C.$,    10,     'ZU,    ,       106,165 ;       U
    344 000076                      CHR     C.$,    10,     'ZV,    ,       106,166 ;       V
    345 000100                      CHR     C.$,    10,     'ZW,    ,       106,167 ;       W
    346 000102                      CHR     C.$,    10,     'ZX,    ,       106,170 ;       X
    347 000104                      CHR     C.$,    10,     'ZY,    ,       106,171 ;       Y
    348             000257          CHRULZ==TMPTAG
    349 000106                      CHR     C.$,    10,     'ZZ,    ,       106,172 ;       Z
    350 000110                      CHR     C.OP,   134,    'QD,    ,       107,114 ;DEL QUAD
    351 000112                      CHR     C.OP,   136,    'GD,    ,       107,115 ;DOWN GRADE
    352 000114                      CHR     C.OP,   140,    'GU,    ,       110,115 ;UP GRADE
    353 000116                      CHR     C.OP,   142,    'QQ,    ,       113,114 ;QUOTE QUAD
    354 000120                      CHR     C.OP,   144,    'RV,    ,       115,117 ;ROTATE
    355 000122                      CHR     C.OP,   54,     'LG,    ,       117,120 ;LOG
    356 000124                      CHR     C.OP,   146,    'CR,    ,       117,137 ;F.C. ROTATE
    357 000126                      CHR     C.OP,   150,    'IQ,    ,       114,133 ;INPUT QUAD
    358 000130                      CHR     C.OP,   152,    'OQ,    ,       114,135 ;OUTPUT QUAD
    359 000132                      CHR     ,       ,       'PD,    ,       107,124 ;PROTECT DEL
    360 000134                      CHR     ,       ,       'XQ,    ,       50,51   ;(FOR PDP10 COMPATABILITY
    361             000273          CHR.OU==TMPTAG
    362 000136                      CHR     ,       ,       'OU,    ,       157,165 ;O<BS>U<BS>T INPUT ESCAPE
    363 000140                      CHR     C.OP,   154,    'PT,    ,       114,131 ;SET FILE POINTER
    364             000160          OP.FMT==160
    365 000142                      CHR     C.OP,   160,    'FM,    ,       112,116 ;FORMAT OPERATOR
```

```
    366 000144                          CHR     C.OP,   112,    ''CO,    ,       106,130 ;UNDERSCORE LEFT U (CONTAINS)
    367 000146                          CHR     C.OP,   116,    ''SS,    ,       106,132 ;UNDERSCORE RIGHT U (SUBSET)
```

TABLES CHARACTER TABLES

```
                000162          OP..AV  ==162                   ;.BXAV - ATOMIC VECTOR
                000164          OP..CT  ==164                   ;.BXCT - COMPARISON TOLERANCE
                000166          OP..IO  ==166                   ;.BXIO - INDEX ORIGIN
                000170          OP..PP  ==170                   ;.BXPP - PRINT PRECISION
                000172          OP..PW  ==172                   ;.BXPW - PRINT WIDTH
                000174          OP..RL  ==174                   ;.BXRL - Random Link
                000176          OP..WA  ==176                   ;.BXWA - WORKSPACE AVAILABLE
                000200          OP..TS  ==200                   ;.BXTS - Time Stamp
                000202          OP..EX  ==202                   ;.BXEX - Expunge
                000204          OP..NC  ==204                   ;.BXNC - Name Classification
                000206          OP..CR  ==206                   ;.BXCR - Canonical Representation
                000210          OP..FX  ==210                   ;.BXFX - Fix function
                000212          OP..LC  ==212                   ;.BXLC - Line Counter
                000214          OP..NL  ==214                   ;.BXNL - Name List

                000300          CHREND==TMPTAG                  ;NUMBER OF CHARACTERS
000043                          .PSECT  TRNTBL                  ;FIX UP SPECIAL ASCII TO APL TRANSLATIONS
000043                                  TRNADR  140,113         ;SPECIAL CASE FOR GRAVE ACCENT
000140  113                             .BYTE 113
000141                                  TRNADR  173,173         ;LEFT BRACE
000173  173                             .BYTE 173
000174                                  TRNADR  174,115         ;RESIDUE
000174  115                             .BYTE 115
000175                                  TRNADR  175,175         ;RIGHT BRACE
000175  175                             .BYTE 175
000176                                  TRNADR  176,124         ;NOT (TILTA)
000176  124                             .BYTE 124
000177                                  TRNADR  177,177         ;FILLER
000177  177                             .BYTE 177

000150                          .PSECT OVRTBL
000150                          OVRTND::                        ;END OF OVRTAB
000262                          .PSECT MNTBL
000262                          MNTBND::                        ;END OF MNTAB
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 52
TABLES          CHARACTER TABLES

      1         000001                          .END
```

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 52-1
SYMBOL TABLE

AC0    =%000000          CHRERR= 104431          CS.BOO= 000010          D.BXIO= 000007          D.$MUL= 000000
AC1    =%000001          CHRFAC= 000217 G        CS.FLO= 000020          D.BXNL= 000007          D.$PLU= 000000
AC2    =%000002          CHRFIL= 000177          CS.IDN= 000040          D.BXPP= 000007          D.$RES= 000000
AC3    =%000003          CHRFRO= 000135          CS.ONE= 000100          D.BXPW= 000007          D.$SUB= 000000
AC4    =%000004          CHRLBR= 000073          CTEPS   000152R 025     D.BXRL= 000007          D.1COM= 000001
AC5    =%000005          CHRLMP= 000224 G        CTMAX   000212R 025     D.CIRC= 000001          D.1EXP= 000002
ALLINT  003274R 025      CHRLPR= 000072          CTRMAX  000172R 025     D.COMB= 000001          D.1ROT= 000003
ARYRET  003276R 025      CHRMNS= 000137          CURPTR= 000200          D.COMP= 000001          EDICHR  005063R 025
ASCII = 000003           CHRMUL= 000075          C.ALOC= 000162          D.CONT= 000006          EDICNT  005064R 025
ASCQUO= 000042           CHRNEG= 000100          C.BLOK= 000064          D.CTLM= 000000          EDIFLG  005062R 025
ASCTAB  000000RG 013     CHRPDL= 000271          C.BLSZ= 000140          D.DEAL= 000003          EDIT   = 000003
A.ALEN= 000004           CHRPLU= 000055          C.BUFF= 177760          D.DECD= 000002          EIS    = 000000
A.ARG = 000040           CHRPWR= 000120          C.BUFL= 177762          D.DIV = 000000          ELESIZ  005132R 025
A.ASB = 000200           CHRQST= 000121          C.CCNT= 177766          D.DMD = 000010          ENDCOR  000004R 025
A.BITS= 000003           CHRQUO= 000113          C.CPTR= 177764          D.DQAD= 177777          EOF    = 104475
A.CON = 000020           CHRRBR= 000047          C.CZ   = 001000         D.DROP= 000002          ERRMAX= 000166
A.DIMS= 000002           CHRRPR= 000042          C.DIG = 000040          D.ENCD= 000002          ERRNUM  004772R 025
A.DIM1= 000006           CHRSEM= 000074          C.FASW= 020000          D.EQ   = 000000         ERRPTR  004770R 025
A.NUM = 000006           CHRSMO= 000112          C.FLEN= 000010          D.EXPD= 000002          ERRO    004774R 025
A.PTR = 000000           CHRSPA= 000040          C.FLSH= 004000          D.FGET= 000005          EXEBUF  005054R 025
A.TMP = 000010           CHRTAB  000000RG 011    C.FRC = 000001          D.FMT = 000004          E$CT    005056R 025
BADDIR= 104463           CHRULA= 000226 G        C.IDEN= 000010          D.FPNT= 000005          E$PT    005060R 025
BADERR= 104437           CHRULZ= 000257 G        C.NAME= 000102          D.FPUT= 000005          FA.APD= 000100
BLZBUB  005116R 025      CHRUPA= 000131          C.NOBF= 000400          D.GE   = 000000         FA.CRE= 000010
BRSELF  005046R 025      CHR.BS= 000010          C.NUM = 000020          D.GT   = 000000         FA.DLK= 001000
CCNT    005072R 025      CHR.CC= 000003          C.OP   = 000004         D.IBM = 000002          FA.ENB= 100000
CHANEX  003060R 025      CHR.CR= 000015          C.OPN = 010000          D.IND = 000002          FA.EXC= 002000
CHAN00  000256R 025      CHR.CZ= 000032          C.RDLN= 177776          D.INNR= 000013          FA.EXT= 000004
CHAN01  000330R 025      CHR.GT= 000046          C.REWT= 002000          D.INTR= 000006          FA.NSP= 000100
CHAN02  000512R 025      CHR.LF= 000012          C.RND = 000010          D.LE   = 000000         FA.NSR= 000200
CHAN03  000674R 025      CHR.LT= 000043          C.SBIT= 177756          D.LOG = 000001          FA.POS= 010000
CHAN04  001056R 025      CHR.OU= 000273 G        C.SPFX= 000100          D.LT   = 000000         FA.RD  = 000001
CHAN05  001240R 025      CHR.SI= 000017          C.SPGL= 000200          D.MAX = 000000          FA.RWD= 004000
CHAN06  001422R 025      CHR.SL= 000057          C.SPLX= 000001          D.MEM = 000002          FA.SEQ= 040000
CHAN07  001604R 025      CHR.SO= 000016          C.SPRS= 000002          D.MIN = 000000          FA.SHR= 000040
CHAN08  001766R 025      CHR.TB= 000011          C.SRC  = 177774         D.MUL = 000000          FA.TMP= 000020
CHAN09  002150R 025      CHR.UL= 000106          C.SRCP= 000024          D.NE   = 000000         FA.WCK= 020000
CHAN10  002332R 025      CHR..A= 000141          C.TMP  = 000061         D.NN   = 000000         FA.WRT= 000002
CHAN11  002514R 025      CHR..E= 000145          C.TT   = 000004         D.NR   = 000000         FD.BLK= 000010
CHAN12  002676R 025      CHR..S= 000163          C.WIDC= 177770          D.OR   = 000000         FD.CCL= 000002
CHAN13  000022R 023      CHR..T= 000164          C.WIDR= 177772          D.PLUS= 000000          FD.COM= 020000
CHAN14  000204R 023      CHR..U= 000165          C.$    = 000011         D.PSUB= 000006          FD.CR  = 000002
CHNOPN= 104471           CHR..Z= 000172          DEBUG  = 000000         D.PWR  = 000001         FD.DIR= 000010
CHRAND= 000051           CHR..0= 000060          DEFERR= 104405          D.QQAD= 000000          FD.FTN= 000001
CHRBEL= 000007           CHR..1= 000061          DEVIL   005114R 025     D.QUAD= 000000          FD.F11= 040000
CHRBKR= 000133           CH.AND= 000001          DEVNTF= 104470          D.RES  = 000000         FD.INS= 000010
CHRBOL= 000200           CMDBUF  003250R 025     DIGITS  000006R 025     D.RHO  = 000000         FD.ISP= 002000
CHRBOX= 000114           CMDMAX= 000010          DIMAX  = 000020         D.ROT  = 000003         FD.MNT= 100000
CHRCMA= 000054           CNTSIZ  005130R 025     DIMBLK  003112R 025     D.SCON= 000006          FD.OSP= 004000
CHRCOL= 000076           COMBIN  005076R 025     DIMPTR  003114R 025     D.SDIF= 000006          FD.PLC= 000004
CHRDEL= 000107           COMCNT  005126R 025     DIMTYP  003116R 025     D.SUB  = 000000         FD.PRN= 000004
CHRDIV= 000053           COMERR= 104425          DIRFUL= 104466          D.SUBS= 000006          FD.PSE= 010000
CHRDLR= 000176           COMPTR  005122R 025     DOMERR= 104417          D.TAKE= 000002          FD.RAH= 000001
CHRDLT= 000110           COMRTN  003156R 025     DPTERR= 104416          D.TRAN= 000004          FD.RAN= 000002
CHRDOT= 000056           COMSKP  005124R 025     DYDCNT  003224R 025     D.UNON= 000006          FD.REC= 000001
CHREND= 000300 G         CORPTR  005120R 025     D.AND  = 000000         D.$MAX= 000000          FD.RWM= 000001
CHREP  = 000105          CS.ASS= 000200          D.BXCT= 000007          D.$MIN= 000000          FD.SDI= 000020
```

SYMBOL TABLE

```
FD.SQD= 000040          F.DVNM= 000134  F.UNIT= 000136      L.BITS= 000003      M.NOT = 000000
FD.TTY= 000004          F.EFBK= 000010  F.URBD= 000020      L.CNT = 000002      M.NUB = 000006
FD.WBH= 000002          F.EFN = 000050  F.VAL = 177776      L.PRNT= 000001      M.NUM = 000204
FF.CHR= 000005          F.END = 002000  F.VBN = 000064      L.PTR = 000000      M.OUTR= 000012
FF.NV = 000003          F.EOBB= 000032  F.VBSZ= 000060      L.PTR1= 000004      M.PITM= 000000
FF.POE= 000002          F.ERR = 000052  GNOME   003102R 025 L.SUBS= 000002      M.PRDS= 000006
FF.RWD= 000001          F.FACC= 000043  HASHC   005136R 025 MACHIN= 000106      M.QQAD= 000000
FF.RWF= 000006          F.FFBY= 000014  HASHTB  005140R 025 MAXCHN= 000014      M.QUAD= 000000
FF.SPC= 000004          F.FLAG= 000002  HSHMSK= 177601      MNTAB   000000RG 016 M.RAND= 000012
FHEADR  005012R   025   F.FLB = 000100  HSHSIZ= 000200      MNTBND  000262RG 016 M.RAVL= 000000
FHRET   005040R   025   F.FNAM= 000110  ICHAN   005050R 025 MNTRTB  000000RG 017 M.RCIP= 000000
FILNTF= 104467          F.FNB = 000102  ILFNAM= 104464      MONBLK  003120R  025 M.RDCT= 000011
FIS   = 000000          F.FNDF= 000010  ILLCHN= 104540      MONCVT  003132R  025 M.REVS= 000003
FISCON= 046000          F.FTYP= 000116  ILLSW = 104565      MONLEN  003130R  025 M.ROLL= 000000
FIXPTR  003074R   025   F.FUN = 000010  INDERR= 104410      MONPTR  003122R  025 M.SATA= 000001
FLAGS   004764R   025   F.FVER= 000120  INNTMP  003250R 025 MONSFT  003126R  025 M.SCAN= 000011
FLCBUF  003250R   025   F.HEAD= 000006  INNTM2  005016R 025 MONTYP  003124R  025 M.SHAP= 000000
FLINE   005014R   025   F.HIBK= 000004  INPTR   000064R 025 MONVAL  003134R  025 M.SIGN= 000000
FLOEXP  003270R   025   F.LBL = 000020  INSNAM= 104462      M.ABS = 000000      M.SUBR= 000001
FLONEG  003272R   025   F.LINE= 000012  IOERR = 104477      M.BOL = 000200      M.SYM = 000203
FLSWTB  000222R   025   F.LOC = 000004  I.CNT = 177776      M.BXAV= 000007      M.SYN = 000201
FLTLEN= 000010          F.LSTC= 000004  I.DIM = 177772      M.BXCR= 000007      M.TRAN= 000012
FNINSI  005044R   025   F.LUN = 000042  I.LEN = 000012      M.BXCT= 000007      M.$ABS= 000000
FORIGN  000032R   025   F.MBCT= 000054  I.OERR= 000062      M.BXEX= 000007      M.$CEI= 000000
FO.APD= 000106          F.MBC1= 000055  I.OLD = 177770      M.BXFX= 000007      M.$FLR= 000000
FO.MFY= 000002          F.MBFG= 000056  I.PTR = 177774      M.BXIO= 000007      M.$IDN= 000000
FO.RD = 000001          F.MON = 000002  I.STR = 177766      M.BXLC= 000003      M.$NEG= 000000
FO.UPD= 000006          F.NIL = 000001  LABERR= 104406      M.BXNC= 000007      M.$SIG= 000000
FO.WRT= 000016          F.NRBD= 000024  LABLPT  005042R 025 M.BXNL= 000007      M.1RDT= 000011
FPMODE  000042R   025   F.NREC= 000030  LARGBL  003162R 025 M.BXPP= 000007      M.1REV= 000003
FPP   = 000000          F.OPN = 000010  LARGCV  003176R 025 M.BXPW= 000007      M.1SCN= 000011
FPPLEN= 000004          F.OVBS= 000030  LARGFL  003160R 025 M.BXRL= 000007      NAMBUF  003240R  025
FREBLK  007400R   025   F.PREV= 000000  LARGLE  003174R 025 M.BXTS= 000007      NB.DEV= 000200
FUZZ    000142R   025   F.PRNT= 100000  LARGLN  003166R 025 M.BXWA= 000007      NB.DIR= 000100
FXPTR1  003076R   025   F.PROT= 010000  LARGPT  003164R 025 M.CEI = 177777      NB.NAM= 000004
F$EVAL= 000100          F.RACC= 000016  LARGSF  003172R 025 M.CHR = 000202      NB.SD1= 000400
F$EVI = 000200          F.RATT= 000001  LARGTY  003170R 025 M.CMT = 000206      NB.SD2= 001000
F$EXEC= 000400          F.RCNM= 000034  LARGVA  003200R 025 M.DOUT= 000006      NB.SNM= 000040
F$FUNC= 001000          F.RCTL= 000017  LBINC   005100R 025 M.DQAD= 000006      NB.STP= 000020
F.ACTL= 000076          F.READ= 004000  LBUFF   003316R 025 M.ENCD= 000004      NB.SVR= 000010
F.ALOC= 000040          F.REDF= 000040  LBUFLN= 000621      M.EXE = 000000      NB.TYP= 000002
F.BBFS= 000062          F.RES = 000040  LCOUNT  005106R 025 M.EXP = 000001      NB.VER= 000001
F.BDB = 000070          F.RET = 177772  LC.KLD= 000000      M.FACT= 000001      ND.DEV= 000200
F.BGBC= 000057          F.RSIZ= 000002  LENERR= 104412      M.FIL = 000177      ND.EXT= 000002
F.BITS= 177774          F.RTYP= 000000  LENTAB  000054R 025 M.FLR = 177777      ND.NAM= 000004
F.BKDN= 000026          F.R2  = 000014  LEXPTR  003100R 025 M.FMT = 000004      ND.PPN= 000100
F.BKDS= 000020          F.R3  = 000016  LFTOP   005066R 025 M.GDUP= 000003      ND.$EX= 000000
F.BKEF= 000050          F.R4  = 000020  LINC    005102R 025 M.GDWN= 000003      ND.$E$= 000020
F.BKP1= 000051          F.SEQN= 000100  LINE    005016R 025 M.GETI= 000006      ND.$NM= 000000
F.BKST= 000024          F.SPDV= 000072  LININC  005026R 025 M.IBM = 000003      ND.$N$= 000040
F.BKVB= 000064          F.SPUN= 000074  LISIZE= 000020      M.IDN = 000000      ND.$PN= 001400
F.BR  = 000020          F.STBK= 000036  LITOP   005072R 025 M.INSP= 000006      NEGSGN  005063R  025
F.CHR = 000075          F.STK = 000022  LNLNG = 104541      M.IOTA= 000000      NINPT1  003302R  025
F.CNTG= 000034          F.STOP= 000004  LOCLCT  005036R 025 M.LIT = 000205      NINPT2  003300R  025
F.CODE= 000010          F.STP = 000001  LOWCOR  000002R 025 M.LN  = 000001      NINSTK  003304R  025
F.DFNB= 000046          F.SUSP= 000002  LSTERP  004766R 025 M.MMD = 000010      NINTMP  003310R  025
F.DSPT= 000044          F.TRC = 000002  LWIDTH  000070R 025 M.NEG = 000000      NINTYP  003306R  025
```

CHRTAB MACRO M1113 08-FEB-80 10:57 PAGE 52-3
SYMBOL TABLE

```
NOTAVL= 104472           OP.FLR= 000014 G        RESBIT  003142R  025 SUBDCT  003106R  025 TTYLEN= 000622
NOTOPN= 104473           OP.FMT= 000160 G        RESCVT  003152R  025 SUBPTR  003110R  025 TURKEY= 104436
NPRT1   004762R  025     OP.MUL= 000006 G        RESERR= 104440           SYMB1   005344R  025 TYPE  = 000166
NULL    007364R  025     OP.NEQ= 000034 G        RESLEN  003150R  025 SYMF    005340R  025 T.BOL  = 020000
NUMCHN= 000020           OP.PLU= 000002 G        RESPTR  003140R  025 SYMPT1  005346R  025 T.CON  = 004000
NUMDFL  003230R  025     OP.RED= 000060 G        RESSFT  003146R  025 SYMTAB  005352R  025 T.DIM  = 000100
NUMEFL  003233R  025     OP.RES= 000012 G        RESTYP  003144R  025 SYMTBF= 104404           T.FUN  = 010000
NUMESN  003236R  025     OP.RHO= 000102 G        RESULT  003136R  025 SYMTOP  005350R  025 T.LIST= 000400
NUMFLG  003230R  025     OP.SCN= 000062 G        RESVAL  003154R  025 SYMTP   006272R  025 T.LIT  = 000020
NUMLEN  000062R  025     OP..AV= 000162 G        RINC    005104R  025 SYMWRN= 000054           T.MON  = 000010
NUMNEG  003232R  025     OP..CR= 000206 G        RLTBAS  005066R  025 SYNERR= 104407           T.NIL  = 000004
NUMQFL  003234R  025     OP..CT= 000164 G        ROTINC  005070R  025 SYSERR= 104402           T.NUM  = 002000
NUMRIG  003231R  025     OP..EX= 000202 G        ROUND   000010R  025 SYSTAT  000000R  024 T.OPFN= 000001
NUMWID  003226R  025     OP..FX= 000210 G        ROVER   003312R  025 SY.BAD= 000001           T.OTHR= 040000
NUMZFL  003235R  025     OP..IO= 000166 G        RSTS  = 000000           SY.BRK= 100000           T.SOP  = 000002
NYIERR= 104432           OP..LC= 000212 G        RSX   = 000115           SY.BS = 000010           T.SPOP= 000040
N.BLSZ= 000036           OP..NC= 000204 G        RTOP    005070R  025 SY.CLR= 000040           T.TMP  = 000200
N.DESC= 000020           OP..NL= 000214 G        RT11  = 000000           SY.ERP= 000020           T.VAL  = 100000
N.DEV = 000032           OP..PP= 000170 G        R.BITS= 000003           SY.NOU= 000100           T.VAR  = 001000
N.DEVN= 000034           OP..PW= 000172 G        R.FIX = 000001           SY.OPN= 000002           UPPER  = 177400
N.DID = 000024           OP..RL= 000174 G        R.LCT = 000002           SY.PMT= 001000           UTNAME  000344R  023
N.DVNM= 000032           OP..TS= 000200 G        R.LN  = 000002           SY.SRT= 000400           VALERR= 104413
N.EXT = 000014           OP..WA= 000176 G        R.LNUM= 000010           SY.SVC= 000004           VARBLE  007400R  025
N.FID = 000000           ORIGIN  000030R  025    R.NEXT= 000006           SY.SVL= 000200           VARBLS  000000R  025
N.FNAM= 000006           OUTPTR  000066R  025    R.OFF = 000000           S.AB  = 177776           VERNUM  000000R  025
N.FTYP= 000014           OVROPR= 000000          R.PREV= 000004           S.ASB = 100000           VRSION= 000002
N.FVER= 000016           OVRTAB  000000RG 014    R.PTR = 000000           S.BFHD= 000020           VTAB    000000RG 012
N.NAME= 000006           OVRTND  000150RG 014    R.SEQ = 000003           S.BITS= 000002           WIDTH   000072R  025
N.NEXT= 000022           OVSTRK= 000214 G        R.VAR = 000002           S.CHR1= 000003           WROUND  000020R  025
N.PPN = 000024           PATCH   007344R  025    R6    =%000006           S.FATT= 000016           WSDATE  000136R  025
N.STAT= 000020           PRECSN  000052R  025    R7    =%000007           S.FDB = 000140           WSFULL= 104403
N.UNIT= 000034           RANDAC  003064R  025    SI    =%000005           S.FLB = 040000           WSNAME  000074R  025
OBUFLN= 001000           RANDY   003062R  025    SIDAMN= 104433           S.FNAM= 000006           WSTIME  000132R  025
OCHAN   005052R  025     RANERR= 104411          SIFRAM  003314R  025 S.FNB = 000036           $FIS  = ****** G
OMODE   000402R  023     RARGBL  003204R  025    SISTAK  006350R  025 S.FNBW= 000017           $FPP  = ****** G
OPTMP0  005066R  025     RARGCV  003220R  025    SITOP   006346R  025 S.FNTY= 000004           $FPPLN= ****** G
OPTMP1  005070R  025     RARGFL  003202R  025    SITP    007144R  025 S.FTYP= 000002           $MACHN= ****** G
OPTMP2  005072R  025     RARGLE  003216R  025    SIWARN= 000200           S.LEN = 000002           $RSTS = ****** G
OPTMP3  005074R  025     RARGLN  003210R  025    SIZTAB  000044R  025 S.LINK= 177774           $RSX  = ****** G
OPTMP4  005076R  025     RARGPT  003206R  025    SI.BIT= 177774           S.NFEN= 000020           $RT11 = ****** G
OPTMP5  005100R  025     RARGSF  003214R  025    SI.LEN= 177772           S.TYPE= 140000           .BOOLE= 000002
OPTMP6  005102R  025     RARGTY  003212R  025    SI.RET= 177772           S.VAL = 000000           .CHARA= 000000
OPTMP7  005104R  025     RARGVA  003222R  025    SI.VAL= 177776           TKDIMS  005072R  025 .DEF. = 000002
OPTMP8  005106R  025     RBASE   005074R  025    SKPSIZ  005134R  025 TKFRAM  005066R  025 .FLOAT= 000006
OPTMP9  005110R  025     RCOUNR  005112R  025    SPHOLD  000004R  024 TKLEFT  005070R  025 .FSRCB= ****** G
OPTM10  005112R  025     RCOUNT  005110R  025    SPINIT  000002R  024 TMODE   000403R  023 .INTEG= 000004
OPTM14  005026R  025     REDCNT  005070R  025    STKINC= 000774           TMPTAG= 000300           .MOLUN  000404R  023
OP.CEI= 000016 G         REDIGN  005066R  025    STOP  = 104420           TRNTAB  000000RG 015 ...GBL= 000000
OP.ENC= 000076 G         REDOFF  005072R  025    SUBCNT  003104R  025 TTYBUF  004140R  025 ...TPC= 000020
OP.EQU= 000030 G         REENTR  003060R  025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
```

M 6

```
CHRTAB  MACRO M1113  08-FEB-80 10:57  PAGE 52-4
SYMBOL TABLE

MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000300      011
VTBL    000300      012
ASCTBL  000600      013
OVRTBL  000150      014
TRNTBL  000200      015
MNTBL   000262      016
MNTRTL  000131      017
CODE    000000      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13748 WORDS  ( 54 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:04:24
CHRTAB,DM1:[305,301]CHRTAB/-SP=APLRX1,APLDEF,CHRTAB,END
```

N 6

FILEID**CKFNSI

```
  CCCCCCCC   KK      KK   FFFFFFFFFF   NN      NN     SSSSSSSS     IIIIII
  CCCCCCCC   KK      KK   FFFFFFFFFF   NN      NN     SSSSSSSS     IIIIII
CC           KK      KK   FF           NN      NN   SS               II
CC           KK      KK   FF           NN      NN   SS               II
CC           KK    KK     FF           NNNN    NN   SS               II
CC           KK    KK     FF           NNNN    NN   SS               II
CC           KKKKKK       FFFFFFFF     NN  NN  NN     SSSSSS         II
CC           KKKKKK       FFFFFFFF     NN  NN  NN     SSSSSS         II
CC           KK    KK     FF           NN    NNNN           SS       II
CC           KK    KK     FF           NN    NNNN           SS       II
CC           KK      KK   FF           NN      NN           SS       II          ....
CC           KK      KK   FF           NN      NN           SS       II          ....
  CCCCCCCC   KK      KK   FF           NN      NN   SSSSSSSS     IIIIII        ....
  CCCCCCCC   KK      KK   FF           NN      NN   SSSSSSSS     IIIIII        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:02
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS                RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CKFNSI           CHECK TO SEE IF THE FUNCTION IS ON THE SI
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  CKFNSI
                                        .IDENT  \V02.03\

                                        .SBTTL  CKFNSI          CHECK TO SEE IF THE FUNCTION IS ON THE SI

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       THIS ROUTINE CHECKS TO SEE WHETHER THE EDITED
                                        ;       FUNCTION IS ON THE SI. IF SO, A ''SI DAMAGE'' MESSAGE
                                        ;       IS ISSUED AND THE SI IS CLEARED.
                                        ;
                                        ; ROUTINES CALLED:
                                        ;       PRINTM  - Prints ''SI DAMAGE'' error message
                                        ;       CLRS1   - Clears the SI, so no deaths can occur
                                        ;--
                                                .PSECT  CODE

                                        CKFNSI::
                                                TST     FNINSI          ;IS THE FUNCTION IN THE SI?
                                                BEQ     1$              ;NO, SKIP THIS MESS
                                                PUSH    <R4,R3,R2,R1,R0>;SAVE A WHOLE BUNCH OF REGISTERS
                                                PUSH    FLAGS           ;SAVE THE FLAG!
                                                QPRINT  DAMNSI          ;DAMN THE SI, FULL SPEED AHEAD!
000024  012701  000000'                        MOV #DAMNSI,R1          ;POINT TO DAMNSI
000034  012737  000040  000000'                MOV     #SY.CLR,SYSTAT  ;CLEAR THE WHOLE THING
000042                                          CALL    CLRS1           ;CLEAR THE SI & RETURN
000046  042716  177623                          BIC     #^C<F$EVAL+F.RE +F.STOP+F.FNDF>,@SP     ;ONLY CERTAIN
000052  052637  004764'                        BIS     (SP)+,FLAGS     ;BITS SHOULD BE SET
000056  005037  005044'                        CLR     FNINSI          ;NO LONGER IN THE SI
000062                                          POP     <R0,R1,R2,R3,R4>;RESTORE THE REGISTERS
000074  062705  000020                        ADD     #20,SI          ;THE LEX ROUTINE PREFERS THIS
000100                                  1$:     RETURN                  ;AND END THIS FOOLISHNESS


000000                                          .PSECT  TABLES

000000                                  DAMNSI:
000000      012     015     163                 .ASCIZ  <12><15>/si damage/<12><15>
000003      151     040     144
000006      141     155     141
000011      147     145     012
000014      015     000
                                                .EVEN
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 46
CKFNSI          CHECK TO SEE IF THE FUNCTION IS ON THE SI

      1         000001                              .END
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.ALOC= 000162    D.CTLM= 000000    EDIFLG  005062R     025
AC1    =%000001          CHRLPR= 000072    C.BLOK= 000064    D.DEAL= 000003    EDIT   = 000003
AC2    =%000002          CHRMNS= 000137    C.BLSZ= 000140    D.DECD= 000002    EIS    = 000000
AC3    =%000003          CHRMUL= 000075    C.BUFF= 177760    D.DIV = 000000    ELESIZ  005132R     025
AC4    =%000004          CHRNEG= 000100    C.BUFL= 177762    D.DMD = 000010    ENDCOR  000004R     025
AC5    =%000005          CHRPDL= 000271    C.CCNT= 177766    D.DQAD= 177777    EOF    = 104475
ALLINT  003274R     025  CHRPLU= 000055    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
ARYRET  003276R     025  CHRPWR= 000120    C.CZ   = 001000   D.ENCD= 000002    ERRNUM  004772R     025
ASCQUO= 000042           CHRQST= 000121    C.DIG  = 000040   D.EQ   = 000000    ERRPTR  004770R     025
A.ALEN= 000004           CHRQUO= 000113    C.FASW= 020000    D.EXPD= 000002    ERR0    004774R     025
A.ARG  = 000040          CHRRBR= 000047    C.FLEN= 000010    D.FGET= 000005    EXEBUF  005054R     025
A.ASB  = 000200          CHRRPR= 000042    C.FLSH= 004000    D.FMT  = 000004   E$CT    005056R     025
A.BITS= 000003           CHRSEM= 000074    C.FRC  = 000001   D.FPNT= 000005    E$PT    005060R     025
A.CON  = 000020          CHRSMO= 000112    C.IDEN= 000010    D.FPUT= 000005    FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040    C.NAME= 000102    D.GE   = 000000   FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131    C.NOBF= 000400    D.GT   = 000000   FA.DLK= 001000
A.NUM  = 000006          CHR.BS= 000010    C.NUM  = 000020   D.IBM  = 000002   FA.ENB= 100000
A.PTR  = 000000          CHR.CC= 000003    C.OP   = 000004   D.IND  = 000002   FA.EXC= 002000
A.TMP  = 000010          CHR.CR= 000015    C.OPN  = 010000   D.INNR= 000013   FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032    C.RDLN= 177776    D.INTR= 000006   FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046    C.REWT= 002000    D.LE   = 000000   FA.NSR= 000200
BLZBUB  005116R     025  CHR.LF= 000012    C.RND  = 000010   D.LOG  = 000001   FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043    C.SBIT= 177756    D.LT   = 000000   FA.RD  = 000001
CCNT    005072R     025  CHR.SI= 000017    C.SPFX= 000100    D.MAX  = 000000   FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057    C.SPGL= 000200    D.MEM  = 000002   FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016    C.SPLX= 000001    D.MIN  = 000000   FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011    C.SPRS= 000002    D.MUL  = 000000   FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106    C.SRC  = 177774   D.NE   = 000000   FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141    C.SRCP= 000024    D.NN   = 000000   FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145    C.TT   = 000004   D.NR   = 000000   FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163    C.WIDC= 177770    D.OR   = 000000   FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164    C.WIDR= 177772    D.PLUS= 000000    FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165    DAMNSI  000000R     002 D.PSUB= 000006    FD.CR  = 000002
CHAN08  001766R     025  CHR..Z= 000172    DEBUG  = 000000   D.PWR  = 000001   FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061    DEVIL   005114R     025 D.QUAD= 000000    FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001    DEVNTF= 104470    D.RES  = 000000   FD.INS= 000010
CHAN12  002676R     025  CKFNSI  000000RG      020 DIGITS  000006R     025 D.RHO  = 000000   FD.ISP= 002000
CHAN13  000022R     023  CLRS1  = ****** GX      DIMAX  = 000020   D.ROT  = 000003   FD.MNT= 100000
CHAN14  000204R     023  CMDBUF  003250R     025 DIMBLK  003112R     025 D.SCON= 000006    FD.OSP= 004000
CHNOPN= 104471           CMDMAX= 000010          DIMPTR  003114R     025 D.SDIF= 000006    FD.PLC= 000004
CHRAND= 000051           CNTSIZ  005130R     025 DIMTYP  003116R     025 D.SUB  = 000000   FD.PRN= 000004
CHRBEL= 000007           COMBIN  005076R     025 DIRFUL= 104466        D.SUBS= 000006    FD.PSE= 010000
CHRBKR= 000133           COMCNT  005126R     025 DOMERR= 104417        D.TAKE= 000002    FD.RAH= 000001
CHRBOL= 000200           COMERR= 104425          DPTERR= 104416        D.TRAN= 000004    FD.RAN= 000002
CHRBOX= 000114           COMPTR  005122R     025 DYDCNT  003224R     025 D.UNON= 000006    FD.REC= 000001
CHRCMA= 000054           COMRTN  003156R     025 D.AND  = 000000       D.$MAX= 000000    FD.RWM= 000001
CHRCOL= 000076           COMSKP  005124R     025 D.BXCT= 000007        D.$MIN= 000000    FD.SDI= 000020
CHRDEL= 000107           CORPTR  005120R     025 D.BXIO= 000007        D.$MUL= 000000    FD.SQD= 000040
CHRDIV= 000053           CS.ASS= 000200          D.BXNL= 000007        D.$PLU= 000000    FD.TTY= 000004
CHRDLR= 000176           CS.BOO= 000010          D.BXPP= 000007        D.$RES= 000000    FD.WBH= 000002
CHRDLT= 000110           CS.FLO= 000020          D.BXPW= 000007        D.$SUB= 000000    FF.CHR= 000005
CHRDOT= 000056           CS.IDN= 000040          D.BXRL= 000007        D.1COM= 000001    FF.NV  = 000003
CHREP  = 000105          CS.ONE= 000100          D.CIRC= 000001        D.1EXP= 000002    FF.POE= 000002
CHRERR= 104431           CTEPS   000152R     025 D.COMB= 000001        D.1ROT= 000003    FF.RWD= 000001
CHRFIL= 000177           CTMAX   000212R     025 D.COMP= 000001        EDICHR  005063R     025 FF.RWF= 000006
CHRFRO= 000135           CTRMAX  000172R     025 D.CONT= 000006        EDICNT  005064R     025 FF.SPC= 000004
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 46-2
SYMBOL TABLE

FHEADR  005012R     025 F.FLB = 000100      HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000
FHRET   005040R     025 F.FNAM= 000110      ICHAN   005050R 025     MONPTR  003122R     025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102      ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010      ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116      ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FIXPTR  003074R     025 F.FUN = 000010      INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R     025 F.FVER= 000120      INNTMP  003250R 025     M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R     025 F.HEAD= 000006      INNTM2  005016R 025     M.BXAV= 000007          M.SYN = 000201
FLINE   005014R     025 F.HIBK= 000004      INPTR   000064R 025     M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R     025 F.LBL = 000020      INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R     025 F.LINE= 000012      IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R     025 F.LOC = 000004      I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004      I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R     025 F.LUN = 000042      I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055      I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056      I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002      I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FO.WRT= 000016          F.NRBD= 000024      LABLPT  005042R 025     M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R     025 F.NREC= 000030      LARGBL  003162R 025     M.BXWA= 000007          NB.DIR= 000100
FPP   = 000000          F.OPN = 000010      LARGCV  003176R 025     M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030      LARGFL  003160R 025     M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R     025 F.PREV= 000000      LARGLE  003174R 025     M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R     025 F.PRNT= 100000      LARGLN  003166R 025     M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R     025 F.PROT= 010000      LARGPT  003164R 025     M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016      LARGSF  003172R 025     M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001      LARGTY  003170R 025     M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034      LARGVA  003200R 025     M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017      LBINC   005100R 025     M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000      LBUFF   003316R 025     M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040      LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040      LCOUNT  005106R 025     M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772      LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002      LENTAB  000054R 025     M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000      LEXPTR  003100R 025     M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014      LFTOP   005066R 025     M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016      LINC    005102R 025     M.IDN = 00000C          ND.$PN= 00140C
F.BKEF= 000050          F.R4  = 000020      LINE    005016R 025     M.INSP= 000006          NEGSGN  005063R     025
F.BKP1= 000051          F.SEQN= 000100      LININC  005026R 025     M.IOTA= 000000          NINPT1  003302R     025
F.BKST= 000024          F.SPDV= 000072      LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.BKVB= 000064          F.SPUN= 000074      LITOP   005072R 025     M.LN  = 000001          NINSTK  003304R     025
F.BR  = 000020          F.STBK= 000036      LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CHR = 000075          F.STK = 000022      LOCLCT  005036R 025     M.NEG = 000000          NINTYP  003306R     025
F.CNTG= 000034          F.STOP= 000004      LOWCOR  000002R 025     M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001      LSTERP  004766R 025     M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002      LWIDTH  000070R 025     M.NUM = 000204          NPRT1   004762R     025
F.DSPT= 000044          F.TRC = 000002      L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.DVNM= 000134          F.UNIT= 000136      L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020      L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EFN = 000050          F.VAL = 177776      L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.END = 002000          F.VBN = 000064      L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.EOBB= 000032          F.VBSZ= 000060      L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.ERR = 000052          GNOME   003102R 025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FACC= 000043          HASHC   005136R 025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025     M.RDCT= 000011          NUMQFL  003234R     025
F.FLAG= 000002          HSHMSK= 177601      MONCVT  003132R 025     M.REVS= 000003          NUMRIG  003231R     025
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 46-3
SYMBOL TABLE

NUMWID  003226R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BS  = 000010          T.LIT  = 000020
NUMZFL  003235R     025 RANERR= 104411          R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
NYIERR= 104432          RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
N.BLSZ= 000036          RARGCV  003220R     025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
N.DESC= 000020          RARGFL  003202R     025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DEV  = 000032         RARGLE  003216R     025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEVN= 000034          RARGLN  003210R     025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DID  = 000024         RARGPT  003206R     025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DVNM= 000032          RARGSF  003214R     025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.EXT  = 000014         RARGTY  003212R     025 SIDAMN= 104433           S.AB   = 177776          T.VAL  = 100000
N.FID  = 000000         RARGVA  003222R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAR  = 001000
N.FNAM= 000006          RBASE   005074R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER  = 177400
N.FTYP= 000014          RCOUNR  005112R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.FVER= 000016          RCOUNT  005110R     025 SITP    007144R     025 S.CHR1= 000003          VALERR= 104413
N.NAME= 000006          REDCNT  005070R     025 SIWARN= 000200           S.FATT= 000016          VARBLE  007400R     025
N.NEXT= 000022          REDIGN  005066R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLS  000000R     025
N.PPN  = 000024         REDOFF  005072R     025 SI.BIT= 177774           S.FLB  = 040000          VERNUM  000000R     025
N.STAT= 000020          REENTR  003060R     025 SI.LEN= 177772           S.FNAM= 000006          VRSION= 000002
N.UNIT= 000034          RESBIT  003142R     025 SI.RET= 177772           S.FNB  = 000036          WIDTH   000072R     025
OBUFLN= 001000          RESCVT  003152R     025 SI.VAL= 177776           S.FNBW= 000017          WROUND  000020R     025
OCHAN   005052R     025 RESERR= 104440          SKPSIZ  005134R     025 S.FNTY= 000004          WSDATE  000136R     025
OMODE   000402R     023 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSFULL= 104403
OPTMP0  005066R     025 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSNAME  000074R     025
OPTMP1  005070R     025 RESSFT  003146R     025 STKINC= 000774           S.LINK= 177774          WSTIME  000132R     025
OPTMP2  005072R     025 RESTYP  003144R     025 STOP   = 104420          S.NFEN= 000020          $FIS   = ****** G
OPTMP3  005074R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FPP   = ****** G
OPTMP4  005076R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPPLN= ****** G
OPTMP5  005100R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP7  005104R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP8  005106R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RT11  = ****** G
OPTMP9  005110R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166              .BOOLE= 000002
OPTM10  005112R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  C04140R     025 .CHARA= 000000
OPTM14  005026R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622              .DEF.  = 000002
ORIGIN  000030R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436              .FLOAT= 000006
OUTPTR  000066R     025 RT11   = 000000          SYMWRN= 000054          TYPE   = 000166              .FSRCB= ****** G
OVROPR= 000000              R.BITS= 000003          SYNERR= 104407          T.BOL  = 020000              .INTEG= 000004
PATCH   007344R     025 R.FIX  = 000001          SYSERR= 104402          T.CON  = 004000              .MOLUN  000404R     023
PRECSN  000052R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100              ...GBL= 000000
PRINTM= ****** GX           R.LN   = 000002          SY.BAD= 000001          T.FUN  = 010000              ...TPC= 000020
RANDAC  003064R     025 R.LNUM= 000010           SY.BRK= 100000          T.LIST= 000400

. ABS.  000000      000
        000000      001
TABLES  000016      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

```
CKFNSI  MACRO M1113  08-FEB-80 11:02  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000     017
CODE    000102     020
PATCH   000000     021
$$FSR1  000000     022
DEF$0   000406     023
DEF$1   000006     024
DEF$2   007400     025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12908 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:49
CKFNSI,DM1:[305,301]CKFNSI/-SP=APLRX1,APLDEF,CKFNSI,END
```

FILEID**CLRCOR

```
  CCCCCCCC    LL            RRRRRRRR        CCCCCCCC     000000     RRRRRRRR
  CCCCCCCC    LL            RRRRRRRR        CCCCCCCC     000000     RRRRRRRR
CC            LL            RR      RR    CC          00      00   RR      RR
CC            LL            RR      RR    CC          00      00   RR      RR
CC            LL            RR      RR    CC          00      00   RR      RR
CC            LL            RR      RR    CC          00      00   RR      RR
CC            LL            RRRRRRRR      CC          00      00   RRRRRRRR
CC            LL            RRRRRRRR      CC          00      00   RRRRRRRR
CC            LL            RR  RR        CC          00      00   RR  RR
CC            LL            RR  RR        CC          00      00   RR  RR
CC            LL            RR    RR      CC          00      00   RR    RR        ....
CC            LL            RR    RR      CC          00      00   RR    RR        ....
  CCCCCCCC    LLLLLLLLLL    RR      RR      CCCCCCCC     000000     RR      RR      ....
  CCCCCCCC    LLLLLLLLLL    RR      RR      CCCCCCCC     000000     RR      RR      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:03
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CLRCOR           CLEAR MEMORY
```

```
                        
     1          000000              RT11    =  0          ;NOT RT-11
     2          000000              RSTS    =  0          ;NOT RSTS/E
     3          000115              RSX     = 'M          ;RSX-11/M
     4
     5          000000              EIS     =  0          ;NO EIS
     6          000000              FIS     =  0          ;NO FIS
     7          000000              FPP     =  0          ;NO FPP
     8
     9          000004              FPPLEN  =  4.         ;SINGLE PRECISION
    10
    11          000000              DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CLRCOR
                                .IDENT  \V02.03\

                                .SBTTL  CLRCOR          CLEAR MEMORY

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is called to initialize
                                ;       the free storage area and the top of core pointer.  The top of core
                                ;       is low enough to accomidate several handlers.  The USR will be swapped
                                ;       in over the pure-code for maximum space.   If APL is running on RSTS
                                ;       the device handlers and the USR are never in user core so no space is
                                ;       left.  The start address (at absolute location 40) is set to 'START',
                                ;       so that when the program is started, this routine won't be called.
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    CLRCOR
                                ;         ..or..
                                ;       CALL    CLRCO1
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       NONE
                                ;
                                ; SIDE EFFECTS;
                                ;       The Workspace is cleared and initialized
                                ;--
000000                          ;       .PSECT  CODE
                                ;
000000                          CLRCO1::
                                ;+
                                ; RESET ALL CHANNELS (Except Overlay Channel)
                                ;-
000000                                  CALL    FRESET          ;RESET CHANNELS 1.-12.
000004  012702  000022'                 MOV     #CHAN13,R2      ;ALSO RESET
000010                                  CALL    RESCHN          ;   UTILTY CHANNEL #1.
000014  012702  000204'                 MOV     #CHAN14,R2      ;ALSO RESET
000020                                  CALL    RESCHN          ;   UTILTY CHANNEL #2.
                                ;
000024                          CLRCOR::
                                ;+
                                ; READ IN A CLEAR CHUCK OF VARIABLE CORE
                                ;-
000024  012700  000040'                 MOV     #CLRSEG,R0      ;POINT TO CLEAR WS SEGMENT
000030                                  CALL    $LOAD           ;FORCE THE DATA INTO CORE
000034  103405                          BCS     14$             ;EVERYTHING WENT WRONG
000036  012700  000046'                 MOV     #ROTSEG,R0      ;Point to Root of 2nd Tree
000042                                  CALL    $LOAD           ;Load it in
000046  103003                          BCC     1$              ;Everything is OK
000050                          14$:
000050                                  EXIT$S                  ;Terminate
000050  012746                          MOV     (PC)+,-(SP)
```

```
CLRCOR  MACRO M1113  08-FEB-80 11:03  PAGE 44-1
CLRCOR          CLEAR MEMORY

        000052     063     001                  .BYTE   51.,1
        000054  104377                          EMT     ^O<377>
    146
    147 000056  042737  000003  000000' 1$:     BIC     #SY.BAD!SY.OPN,SYSTAT   ;WS no longer Possibly Bad
```

```
CLRCOR  MACRO M1113  08-FEB-80 11:03  PAGE 45
CLRCOR           CLEAR MEMORY


                                        ;+
                                        ; NOW TO INITIALIZE WHAT WE READ IN
                                        ;-
000064  162706  000040                  SUB     #32.,SP         ;SPACE FOR PARTITION PARAMETERS
000070  010601                          MOV     SP,R1           ;POINTER IN R1
000072                                  GTSK$S  R1              ;GET THE PARAMETERS
000072  010146                          MOV     R1,-(SP)
000074  012746                          MOV     (PC)+,-(SP)
000076     077     002                  .BYTE   63.,2
000100  104377                          EMT     ^O<377>
000102  016100  000032                  MOV     G.TSTS(R1),R0   ;GET THE SIZE
000106  062706  000040                  ADD     #32.,SP         ;SKIP THE GARBAGE
000112  162700  000002                  SUB     #2.,R0          ;MINUS THE HIGH LOCATION
000116  042700  000001                  BIC     #1.,R0          ;JUST IN CASE
000122  012701  007376'                 MOV     #FREBLK-2,R1    ;Get Pointer to Low Core-2
000126  012721  177777                  MOV     #-1,(R1)+       ;PUT IN SAFETY WORD
000132  010037  000004'                 MOV     R0,ENDCOR       ;SET A POINTER TO TOP OF APL CORE
000136  010137  000002'                 MOV     R1,LOWCOR       ;LOWER CORE LIMIT
000142  160100                          SUB     R1,R0           ;GET LENGTH OF FREE BLOCK
000144  006200                          ASR     R0              ;MAKE A WORD SIZE FOR THE BLOCK
000146  010021                          MOV     R0,(R1)+        ;SAVE AS SIZE WORD IN BLOCK
000150  010137  003312'                 MOV     R1,ROVER        ;SET POINTER TO NEXT (ONLY) FREE BLOCK
000154  010111                          MOV     R1,(R1)         ;PUT IN LINKS
000156  010161  000002                  MOV     R1,2(R1)        ;BOTH OF THEM
000162  060001                          ADD     R0,R1           ;MOVE TO END OF BLOCK
000164  060001                          ADD     R0,R1           ;DID THIS TWICE BECAUSE IT'S WORD COUNT
000166  012741  177777                  MOV     #-1,-(R1)       ;SET UP SAFETY WORD
000172  010041                          MOV     R0,-(R1)        ;PUT IN LENGTH AT END
000174  105737  000402'                 TSTB    OMODE           ;OMODE already set?
000200  002002                          BGE     2$              ;Yes, don't initialize
000202                                  CALL    TTINIT          ;Initialize the terminal
000206                          2$:     RETURN                  ;And return


000000                                  .PSECT  TABLES

000000                          CHANNEL::
000000  000256' 000330' 000512'         .WORD   CHAN00, CHAN01, CHAN02, CHAN03, CHAN04
000006  000674' 001056'
000012  001240' 001422' 001604'         .WORD   CHAN05, CHAN06, CHAN07, CHAN08, CHAN09
000020  001766' 002150'
000024  002332' 002514' 002676'         .WORD   CHAN10, CHAN11, CHAN12, CHAN13, CHAN14
000032  000022' 000204'
000036  003060'                         .WORD   CHANEX
000040     003     000          CLRSEG: .BYTE   3,0             ;LOAD SYNCHRONOUSLY
000042  004314  015020                  .RAD50  /APLDF2/        ;THE VARIABLES (CLEAR IMAGE)

000046     003     000          ROTSEG: .BYTE   3,0             ;LOAD SYNCHRONOUSLY
000050  071347  100730                  .RAD50  /ROOT1 /        ;    the 2nd tree root
```

```
CLRCOR  MACRO M1113  08-FEB-80 11:03  PAGE 47
CLRCOR           CLEAR MEMORY

      1          000001                              .END
```

CLRCOR MACRO M1113 08-FEB-80 11:03 PAGE 47-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135          CTMAX   000212R  025 D.CONT= 000006          EDICNT  005064R  025
AC1    =%000001          CHRLBR= 000073          CTRMAX  000172R  025 D.CTLM= 000000          EDIFLG  005062R  025
AC2    =%000002          CHRLPR= 000072          C.ALOC= 000162      D.DEAL= 000003          EDIT   = 000003
AC3    =%000003          CHRMNS= 000137          C.BLOK= 000064      D.DECD= 000002          EIS    = 000000
AC4    =%000004          CHRMUL= 000075          C.BLSZ= 000140      D.DIV = 000000          ELESIZ  005132R  025
AC5    =%000005          CHRNEG= 000100          C.BUFF= 177760      D.DMD = 000010          ENDCOR  000004R  025
ALLINT  003274R  025 CHRPDL= 000271          C.BUFL= 177762      D.DQAD= 177777          EOF    = 104475
ARYRET  003276R  025 CHRPLU= 000055          C.CCNT= 177766      D.DROP= 000002          ERRMAX= 000166
ASCQUO= 000042          CHRPWR= 000120          C.CPTR= 177764      D.ENCD= 000002          ERRNUM  004772R  025
A.ALEN= 000004          CHRQST= 000121          C.CZ   = 001000     D.EQ   = 000000         ERRPTR  004770R  025
A.ARG = 000040          CHRQUO= 000113          C.DIG = 000040      D.EXPD= 000002          ERR0    004774R  025
A.ASB = 000200          CHRRBR= 000047          C.FASW= 020000      D.FGET= 000005          EXEBUF  005054R  025
A.BITS= 000003          CHRRPR= 000042          C.FLEN= 000010      D.FMT = 000004          E$CT    005056R  025
A.CON = 000020          CHRSEM= 000074          C.FLSH= 004000      D.FPNT= 000005          E$PT    005060R  025
A.DIMS= 000002          CHRSMO= 000112          C.FRC = 000001      D.FPUT= 000005          FA.APD= 000100
A.DIM1= 000006          CHRSPA= 000040          C.IDEN= 000010      D.GE   = 000000         FA.CRE= 000010
A.NUM = 000006          CHRUPA= 000131          C.NAME= 000102      D.GT   = 000000         FA.DLK= 001000
A.PTR = 000000          CHR.BS= 000010          C.NOBF= 000400      D.IBM = 000002          FA.ENB= 100000
A.TMP = 000010          CHR.CC= 000003          C.NUM = 000020      D.IND = 000002          FA.EXC= 002000
BADDIR= 104463          CHR.CR= 000015          C.OP   = 000004     D.INNR= 000013          FA.EXT= 000004
BADERR= 104437          CHR.CZ= 000032          C.OPN = 010000      D.INTR= 000006          FA.NSP= 000100
BLZBUB  005116R  025 CHR.GT= 000046          C.RDLN= 177776      D.LE   = 000000         FA.NSR= 000200
BRSELF  005046R  025 CHR.LF= 000012          C.REWT= 002000      D.LOG = 000001          FA.POS= 010000
CCNT    005072R  025 CHR.LT= 000043          C.RND = 000010      D.LT   = 000000         FA.RD  = 000001
CHANEX  003060R  025 CHR.SI= 000017          C.SBIT= 177756      D.MAX = 000000          FA.RWD= 004000
CHANNE  000000RG  002 CHR.SL= 000057          C.SPFX= 000100      D.MEM = 000002          FA.SEQ= 040000
CHAN00  000256R  025 CHR.SO= 000016          C.SPGL= 000200      D.MIN = 000000          FA.SHR= 000040
CHAN01  000330R  025 CHR.TB= 000011          C.SPLX= 000001      D.MUL = 000000          FA.TMP= 000020
CHAN02  000512R  025 CHR.UL= 000106          C.SPRS= 000002      D.NE   = 000000         FA.WCK= 020000
CHAN03  000674R  025 CHR..A= 000141          C.SRC  = 177774     D.NN   = 000000         FA.WRT= 000002
CHAN04  001056R  025 CHR..E= 000145          C.SRCP= 000024      D.NR   = 000000         FD.BLK= 000010
CHAN05  001240R  025 CHR..S= 000163          C.TT   = 000004     D.OR   = 000000         FD.CCL= 000002
CHAN06  001422R  025 CHR..T= 000164          C.WIDC= 177770      D.PLUS= 000000          FD.COM= 020000
CHAN07  001604R  025 CHR..U= 000165          C.WIDR= 177772      D.PSUB= 000006          FD.CR  = 000002
CHAN08  001766R  025 CHR..Z= 000172          DEBUG  = 000000     D.PWR  = 000001         FD.DIR= 000010
CHAN09  002150R  025 CHR..0= 000060          DEFERR= 104405      D.QQAD= 000000          FD.FTN= 000001
CHAN10  002332R  025 CHR..1= 000061          DEVIL   005114R  025 D.QUAD= 000000          FD.F11= 040000
CHAN11  002514R  025 CH.AND= 000001          DEVNTF= 104470      D.RES  = 000000         FD.INS= 000010
CHAN12  002676R  025 CLRCOR  000024RG  020 DIGITS  000006R  025 D.RHO  = 000000         FD.ISP= 002000
CHAN13  000022R  023 CLRCO1  000000RG  020 DIMAX  = 000020      D.ROT  = 000003         FD.MNT= 100000
CHAN14  000204R  023 CLRSEG  000040R  002 DIMBLK  003112R  025 D.SCON= 000006          FD.OSP= 004000
CHNOPN= 104471          CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006          FD.PLC= 000004
CHRAND= 000051          CMDMAX= 000010      DIMTYP  003116R  025 D.SUB  = 000000         FD.PRN= 000004
CHRBEL= 000007          CNTSIZ  005130R  025 DIRFUL= 104466      D.SUBS= 000006          FD.PSE= 010000
CHRBKR= 000133          COMBIN  005076R  025 DOMERR= 104417      D.TAKE= 000002          FD.RAH= 000001
CHRBOL= 000200          COMCNT  005126R  025 DPTERR= 104416      D.TRAN= 000004          FD.RAN= 000002
CHRBOX= 000114          COMERR= 104425      DYDCNT  003224R  025 D.UNON= 000006          FD.REC= 000001
CHRCMA= 000054          COMPTR  005122R  025 D.AND  = 000000      D.$MAX= 000000          FD.RWM= 000001
CHRCOL= 000076          COMRTN  003156R  025 D.BXCT= 000007      D.$MIN= 000000          FD.SDI= 000020
CHRDEL= 000107          COMSKP  005124R  025 D.BXIO= 000007      D.$MUL= 000000          FD.SQD= 000040
CHRDIV= 000053          CORPTR  005120R  025 D.BXNL= 000007      D.$PLU= 000000          FD.TTY= 000004
CHRDLR= 000176          CS.ASS= 000200      D.BXPP= 000007      D.$RES= 000000          FD.WBH= 000002
CHRDLT= 000110          CS.BOO= 000010      D.BXPW= 000007      D.$SUB= 000000          FF.CHR= 000005
CHRDOT= 000056          CS.FLO= 000020      D.BXRL= 000007      D.1COM= 000001          FF.NV  = 000003
CHREP  = 000105         CS.IDN= 000040      D.CIRC= 000001      D.1EXP= 000002          FF.POE= 000002
CHRERR= 104431          CS.ONE= 000100      D.COMB= 000001      D.1ROT= 000003          FF.RWD= 000001
CHRFIL= 000177          CTEPS   000152R  025 D.COMP= 000001      EDICHR  005063R  025 FF.RWF= 000006
```

```
CLRCOR  MACRO M1113  08-FEB-80 11:03  PAGE 47-2
SYMBOL TABLE

FF.SPC= 000004          F.FFBY= 000014          G.TSFW= 000024          LITOP   005072R 025 M.LN  = 000001
FHEADR  005012R 025     F.FLAG= 000002          G.TSGC= 000017          LNLNG = 104541      M.MMD = 000010
FHRET   005040R 025     F.FLB = 000100          G.TSMT= 000022          LOCLCT  005036R 025 M.NEG = 000000
FILNTF= 104467          F.FNAM= 000110          G.TSNL= 000020          LOWCOR  000002R 025 M.NOT = 000000
FIS   = 000000          F.FNB = 000102          G.TSPC= 000016          LSTERP  004766R 025 M.NUB = 000006
FISCON= 046000          F.FNDF= 000010          G.TSPN= 000004          LWIDTH  000070R 025 M.NUM = 000204
FIXPTR  003074R 025     F.FTYP= 000116          G.TSPR= 000014          L.BITS= 000003      M.OUTR= 000012
FLAGS   004764R 025     F.FUN = 000010          G.TSRN= 000010          L.CNT = 000002      M.PITM= 000000
FLCBUF  003250R 025     F.FVER= 000120          G.TSSY= 000034          L.PRNT= 000001      M.PRDS= 000006
FLINE   005014R 025     F.HEAD= 000006          G.TSTN= 000000          L.PTR = 000000      M.QQAD= 000000
FLOEXP  003270R 025     F.HIBK= 000004          G.TSTS= 000032          L.PTR1= 000004      M.QUAD= 000000
FLONEG  003272R 025     F.LBL = 000020          G.TSVA= 000026          L.SUBS= 000002      M.RAND= 000012
FLSWTB  000222R 025     F.LINE= 000012          G.TSVL= 000030          MACHIN= 000106      M.RAVL= 000000
FLTLEN= 000010          F.LOC = 000004          HASHC   005136R 025 MAXCHN= 000014          M.RCIP= 000000
FNINSI  005044R 025     F.LSTC= 000004          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011
FORIGN  000032R 025     F.LUN = 000042          HSHMSK= 177601      MONCVT  003132R 025 M.REVS= 000003
FO.APD= 000106          F.MBCT= 000054          HSHSIZ= 000200      MONLEN  003130R 025 M.ROLL= 000000
FO.MFY= 000002          F.MBC1= 000055          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FO.RD = 000001          F.MBFG= 000056          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FO.UPD= 000006          F.MON = 000002          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FO.WRT= 000016          F.NIL = 000001          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FPMODE  000042R 025     F.NRBD= 000024          INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FPP   = 000000          F.NREC= 000030          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FPPLEN= 000004          F.OPN = 000010          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FREBLK  007400R 025     F.OVBS= 000030          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FRESET= ****** GX       F.PREV= 000000          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FUZZ    000142R 025     F.PRNT= 100000          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FXPTR1  003076R 025     F.PROT= 010000          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
F$EVAL= 000100          F.RACC= 000016          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
F$EVI = 000200          F.RATT= 000001          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
F$EXEC= 000400          F.RCNM= 000034          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
F$FUNC= 001000          F.RCTL= 000017          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
F.ACTL= 000076          F.READ= 004000          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
F.ALOC= 000040          F.REDF= 000040          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
F.BBFS= 000062          F.RES = 000040          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
F.BDB = 000070          F.RET = 177772          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
F.BGBC= 000057          F.RSIZ= 000002          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
F.BITS= 177774          F.RTYP= 000000          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
F.BKDN= 000026          F.R2  = 000014          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
F.BKDS= 000020          F.R3  = 000016          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
F.BKEF= 000050          F.R4  = 000020          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
F.BKP1= 000051          F.SEQN= 000^O           LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F.BKST= 000024          F.SPDV= 000072          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F.BKVB= 000064          F.SPUN= 000074          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F.BR  = 000020          F.STBK= 000036          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F.CHR = 000075          F.STK = 000022          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.CNTG= 000034          F.STOP= 000004          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.CODE= 000010          F.STP = 000001          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.DFNB= 000046          F.SUSP= 000002          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.DSPT= 000044          F.TRC = 000002          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.DVNM= 000134          F.UNIT= 000136          LENTAB  000054R 025 M.GDWN= 000003      ND.$E$= 000020
F.EFBK= 000010          F.URBD= 000020          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.EFN = 000050          F.VAL = 177776          LFTOP   005066R 025 M.IBM = 000003      ND.$N$= 000040
F.END = 002000          F.VBN = 000064          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.EOBB= 000032          F.VBSZ= 000060          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.ERR = 000052          GNOME   003102R 025     LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.FACC= 000043          G.TSDU= 000036          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
```

CLRCOR MACRO M1113 08-FEB-80 11:03 PAGE 47-3
SYMBOL TABLE

```
NINSTK  003304R     025 OPTMP8  005106R     025 RTOP    005070R     025 SYMWRN= 000054          T.DIM = 000100
NINTMP  003310R     025 OPTMP9  005110R     025 RT11  = 000000          SYNERR= 104407          T.FUN = 010000
NINTYP  003306R     025 OPTM10  005112R     025 RX$IAS= 000003          SYSERR= 104402          T.LIST= 000400
NOTAVL= 104472          OPTM14  005026R     025 RX$11D= 000000          SYSTAT  000000R     024 T.LIT = 000020
NOTOPN= 104473          ORIGIN  000030R     025 RX$11M= 000001          SY.BAD= 000001          T.MON = 000010
NPRT1   004762R     025 OUTPTR  000066R     025 RX$11S= 000002          SY.BRK= 100000          T.NIL = 000004
NULL    007364R     025 OVROPR= 000000          R.BITS= 000003          SY.BS = 000010          T.NUM = 002000
NUMCHN= 000020          PATCH   007344R     025 R.FIX = 000001          SY.CLR= 000040          T.OPFN= 000001
NUMDFL  003230R     025 PRECSN  000052R     025 R.LCT = 000002          SY.ERP= 000020          T.OTHR= 040000
NUMEFL  003233R     025 RANDAC  003064R     025 R.LN  = 000002          SY.NOU= 000100          T.SOP = 000002
NUMESN  003236R     025 RANDY   003062R     025 R.LNUM= 000010          SY.OPN= 000002          T.SPOP= 000040
NUMFLG  003230R     025 RANERR= 104411          R.NEXT= 000006          SY.PMT= 001000          T.TMP = 000200
NUMLEN  000062R     025 RARGBL  003204R     025 R.OFF = 000000          SY.SRT= 000400          T.VAL = 100000
NUMNEG  003232R     025 RARGCV  003220R     025 R.PREV= 000004          SY.SVC= 000004          T.VAR = 001000
NUMQFL  003234R     025 RARGFL  003202R     025 R.PTR = 000000          SY.SVL= 000200          UPPER = 177400
NUMRIG  003231R     025 RARGLE  003216R     025 R.SEQ = 000003          S.AB  = 177776          UTNAME  000344R     023
NUMWID  003226R     025 RARGLN  003210R     025 R.VAR = 000002          S.ASB = 100000          VALERR= 104413
NUMZFL  003235R     025 RARGPT  003206R     025 R6    =%000006          S.BFHD= 000020          VARBLE  007400R     025
NYIERR= 104432          RARGSF  003214R     025 R7    =%000007          S.BITS= 000002          VARBLS  000000R     025
N.BLSZ= 000036          RARGTY  003212R     025 SI    =%000005          S.CHR1= 000003          VERNUM  000000R     025
N.DESC= 000020          RARGVA  003222R     025 SIDAMN= 104433          S.FATT= 000016          VRSION= 000002
N.DEV = 000032          RBASE   005074R     025 SIFRAM  003314R     025 S.FDB = 000140          WIDTH   000072R     025
N.DEVN= 000034          RCOUNR  005112R     025 SISTAK  006350R     025 S.FLB = 040000          WROUND  000020R     025
N.DID = 000024          RCOUNT  005110R     025 SITOP   006346R     025 S.FNAM= 000006          WSDATE  000136R     025
N.DVNM= 000032          REDCNT  005070R     025 SITP    007144R     025 S.FNB = 000036          WSFULL= 104403
N.EXT = 000014          REDIGN  005066R     025 SIWARN= 000200          S.FNBW= 000017          WSNAME  000074R     025
N.FID = 000000          REDOFF  005072R     025 SIZTAB  000044R     025 S.FNTY= 000004          WSTIME  000132R     025
N.FNAM= 000006          REENTR  003060R     025 SI.BIT= 177774          S.FTYP= 000002          $FIS  = ****** G
N.FTYP= 000014          RESBIT  003142R     025 SI.LEN= 177772          S.LEN = 000002          $FPP  = ****** G
N.FVER= 000016          RESCHN= ****** GX       SI.RET= 177772          S.LINK= 177774          $FPPLN= ****** G
N.NAME= 000006          RESCVT  003152R     025 SI.VAL= 177776          S.NFEN= 000020          $LOAD = ****** GX
N.NEXT= 000022          RESERR= 104440          SKPSIZ  005134R     025 S.TYPE= 140000          $MACHN= ****** G
N.PPN = 000024          RESLEN  003150R     025 SPHOLD  000004R     024 S.VAL = 000000          $RSTS = ****** G
N.STAT= 000020          RESPTR  003140R     025 SPINIT  000002R     024 TKDIMS  005072R     025 $RSX  = ****** G
N.UNIT= 000034          RESSFT  003146R     025 STKINC= 000774          TKFRAM  005066R     025 $RT11 = ****** G
OBUFLN= 001000          RESTYP  003144R     025 STOP  = 104420          TKLEFT  005070R     025 $$$OST= 000004
OCHAN   005052R     025 RESULT  003136R     025 SUBCNT  003104R     025 TMODE   000403R     023 .BOOLE= 000002
OMODE   000402R     023 RESVAL  003154R     025 SUBDCT  003106R     025 TMPTAG= 000166          .CHARA= 000000
OPTMP0  005066R     025 RINC    005104R     025 SUBPTR  003110R     025 TTINIT= ****** GX       .DEF. = 000002
OPTMP1  005070R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TTYBUF  004140R     025 .FLOAT= 000006
OPTMP2  005072R     025 ROTINC  005070R     025 SYMF    005340R     025 TTYLEN= 000622          .FSRCB= ****** G
OPTMP3  005074R     025 ROTSEG  000046R     002 SYMPT1  005346R     025 TURKEY= 104436          .INTEG= 000004
OPTMP4  005076R     025 ROUND   000010R     025 SYMTAB  005352R     025 TYPE  = 000166          .MOLUN  000404R     023
OPTMP5  005100R     025 ROVER   003312R     025 SYMTBF= 104404          T.BOL = 020000          ...GBL= 000000
OPTMP6  005102R     025 RSTS  = 000000          SYMTOP  005350R     025 T.CON = 004000          ...TPC= 000020
OPTMP7  005104R     025 RSX   = 000115          SYMTP   006272R     025

. ABS.  000000      000
        000000      001
TABLES  000054      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
```

K 8

```
CLRCOR  MACRO M1113  08-FEB-80 11:03  PAGE 47-4
SYMBOL TABLE

VTBL     000000      012
ASCTBL   000000      013
OVRTBL   000000      014
TRNTBL   000000      015
MNTBL    000000      016
MNTRTL   000000      017
CODE     000210      020
PATCH    000000      021
$$FSR1   000000      022
DEF$0    000406      023
DEF$1    000006      024
DEF$2    007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13202 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:10
CLRCOR,DM1:[305,301]CLRCOR/-SP=APLRX1,APLDEF,CLRCOR,END
```

FILEID**CLRSI

```
  CCCCCCCC   LL          RRRRRRRR       SSSSSSSS     IIIIII
  CCCCCCCC   LL          RRRRRRRR       SSSSSSSS     IIIIII
CC           LL          RR      RR   SS             II
CC           LL          RR      RR   SS             II
CC           LL          RR      RR   SS             II
CC           LL          RR      RR   SS             II
CC           LL          RRRRRRRR       SSSSSS       II
CC           LL          RRRRRRRR       SSSSSS       II
CC           LL          RR  RR                SS     II
CC           LL          RR  RR                SS     II
CC           LL          RR    RR              SS     II          ....
CC           LL          RR    RR              SS     II          ....
  CCCCCCCC   LLLLLLLLLL  RR      RR   SSSSSSSS     IIIIII        ....
  CCCCCCCC   LLLLLLLLLL  RR      RR   SSSSSSSS     IIIIII        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:06
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CLRSI            CLEAR STATE INDICATOR
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 1


      1         000000                RT11    =  0          ;NOT RT-11
      2         000000                RSTS    =  0          ;NOT RSTS/E
      3         000115                RSX     =  'M         ;RSX-11/M
      4
      5         000000                EIS     =  0          ;NO EIS
      6         000000                FIS     =  0          ;NO FIS
      7         000000                FPP     =  0          ;NO FPP
      8
      9         000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11         000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                        .TITLE  CLRSI
                        .IDENT  \V02.03\

                        .SBTTL  CLRSI           CLEAR STATE INDICATOR

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       'CLRSI' is invoked when a single right arrow
                        ;       is found on a line.  This routine kills
                        ;       evaluation and function frames on the state
                        ;       indicator until it comes to an execution frame.
                        ;       Thus, this routine will clear the state indicator
                        ;       stack back to the last suspended function or to
                        ;       the beginning, whichever happens first
                        ;
                        ; CALLING SEQUENCE:
                        ;       To clear SI, spawn SI frame:
                        ;               JMP     CLRSI
                        ;       To clear SI, return (for branch)
                        ;               CALL    CLRS1
                        ;
                        ; INPUTS:
                        ;       SYSTAT  - SY.CLR indicates whether to clear the entire SI
                        ;                 or just to previous frame.
                        ;
                        ; OUTPUTS:
                        ;       NONE
                        ;
                        ; ROUTINES CALLED:
                        ;       EXEC    - Where to go to spawn new frame
                        ;       FNEXIT  - Clean up after Function
                        ;       KILL    - Delete args on SI
                        ;       PKILL   - Delete (multiple) args on SI
                        ;--


000000                  .PSECT  CODE

000000  013706  000002' CLRSI:: MOV     SPINIT,SP       ;RE-INITIALIZE THE STACK
000004                  PUSH    #EXEC           ;'RETURN' ADDRESS
000010                  CLRS1::
000010  013703  003314' MOV     SIFRAM,R3       ;POINT TO TOP FRAME
000014  032737  000100  004764' BIT     #F$EVAL,FLAGS   ;E-FRAME?
000022  001402          BEQ     1$              ;NO, FIND OUT WHAT TYPE OF FRAME
000024  005037  004764' CLR     FLAGS           ;FUDGE SO WE DON'T STOP ON THIS FRAME

000030                  1$:     CALL    PKILL           ;KILL ALL PARSE FRAMES

000034  032737  000100  004764' 2$:     BIT     #F$EVAL,FLAGS   ;EVAL FRAME?
000042  001405          BEQ     3$              ;NO, KEEP LOOKING
000044  032737  000040  000000' BIT     #SY.CLR,SYSTAT  ;CLEARING ENTIRE SI STACK?
000052  001017          BNE     4$              ;YES, SO KEEP CLEARING
000054                  RETURN                  ;YES, DONE WITH THIS PORTION

000056  032737  000600  004764' 3$:     BIT     #F$EXEC+F$EVI,FLAGS     ;EXECUTION OR EVI FRAME?
000064  001412          BEQ     4$              ;NOPE
000066  013705  003314' MOV     SIFRAM,SI       ;YES, SO INPUT POINTERS MUST BE RESET
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 44-1
CLRSI           CLEAR STATE INDICATOR

     58 000072  016505  000006                    MOV     F.HEAD(SI),SI   ;POINT PAST HEADER DATA
     59 000076  014537  005056'                   MOV     -(SI),E$CT      ;RESTORE OLD COUNT
     60 000102  014537  005060'                   MOV     -(SI),E$PT      ;AND POINTER
     61 000106  014537  000064'                   MOV     -(SI),INPTR     ;MOST INPORTANT!! OLD INPUT POINTER
     62
     63 000112  032737  001000  004764' 4$:      BIT     #F$FUNC,FLAGS   ;IS IT A FUNCTION?
     64 000120  001407                            BEQ     5$              ;NO
     65 000122                                    CALL    FNEXIT          ;YES, POP FUNCTION FRAME
     66 000126  062706  000010                    ADD     #8.,SP          ;SKIP OVER RESTORED STUFF
     67 000132                                    CALL    KILL            ;KILL RESULT OF FUNCTION
     68 000136  000734                            BR      1$              ;AND CYCLE
     69
     70 000140  013705  003314'          5$:      MOV     SIFRAM,SI       ;POINT TO REF. PT. OF FRAME
     71 000144  011503                            MOV     @SI,R3          ;POINT TO PREVIOUS FRAME
     72 000146  001406                            BEQ     6$              ;END OF SI STACK!
     73 000150  010337  003314'                   MOV     R3,SIFRAM       ;SAVE PREVIOUS FRAME POINTER
     74 000154  016337  000002  004764'          MOV     F.FLAG(R3),FLAGS;POP PREVIOUS FRAME'S FLAGS
     75 000162  000722                            BR      1$              ;CONTINUE
     76
     77 000164  042737  000040  000000' 6$:      BIC     #SY.CLR,SYSTAT  ;CLEAR CLEAR SI FLAG
     78 000172  012737  000100  004764'          MOV     #F$EVAL,FLAGS   ;SET FLAGS
     79 000200  005065  177772                    CLR     F.RET(SI)       ;OH, HE NEVER RETURNED, NO HE...
     80 000204  010500                            MOV     SI,R0           ;COPY SI
     81 000206  062700  000020                    ADD     #F.CODE+10,R0   ;ADD SIZE
     82 000212  010065  000006                    MOV     R0,F.HEAD(SI)   ;AND POINT AFTER IT ALL
     83 000216                                    RETURN                  ;AND RETURN
     84
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 46
CLRSI           CLEAR STATE INDICATOR

      1         000001                          .END
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.ALOC= 000162    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLPR= 000072    C.BLOK= 000064    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRMNS= 000137    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMUL= 000075    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRNEG= 000100    C.BUFL= 177762    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRPDL= 000271    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPLU= 000055    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPWR= 000120    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRQST= 000121    C.DIG = 000040    D.EXPD= 000002    ERR0    004774R     025
A.ALEN= 000004           CHRQUO= 000113    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRRBR= 000047    C.FLEN= 000010    D.FMT = 000004    EXEC  = ****** GX
A.ASB = 000200           CHRRPR= 000042    C.FLSH= 004000    D.FPNT= 000005    E$CT    005056R     025
A.BITS= 000003           CHRSEM= 000074    C.FRC = 000001    D.FPUT= 000005    E$PT    005060R     025
A.CON = 000020           CHRSMO= 000112    C.IDEN= 000010    D.GE  = 000000    FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040    C.NAME= 000102    D.GT  = 000000    FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131    C.NOBF= 000400    D.IBM = 000002    FA.DLK= 001000
A.NUM = 000006           CHR.BS= 000010    C.NUM = 000020    D.IND = 000002    FA.ENB= 100000
A.PTR = 000000           CHR.CC= 000003    C.OP  = 000004    D.INNR= 000013    FA.EXC= 002000
A.TMP = 000010           CHR.CR= 000015    C.OPN = 010000    D.INTR= 000006    FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032    C.RDLN= 177776    D.LE  = 000000    FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046    C.REWT= 002000    D.LOG = 000001    FA.NSR= 000200
BLZBUB  005116R     025  CHR.LF= 000012    C.RND = 000010    D.LT  = 000000    FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043    C.SBIT= 177756    D.MAX = 000000    FA.RD = 000001
CCNT    005072R     025  CHR.SI= 000017    C.SPFX= 000100    D.MEM = 000002    FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057    C.SPGL= 000200    D.MIN = 000000    FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016    C.SPLX= 000001    D.MUL = 000000    FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011    C.SPRS= 000002    D.NE  = 000000    FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106    C.SRC = 177774    D.NN  = 000000    FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141    C.SRCP= 000024    D.NR  = 000000    FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145    C.TT  = 000004    D.OR  = 000000    FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163    C.WIDC= 177770    D.PLUS= 000000    FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164    C.WIDR= 177772    D.PSUB= 000006    FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165    DEBUG = 000000    D.PWR = 000001    FD.CR = 000002
CHAN08  001766R     025  CHR..Z= 000172    DEFERR= 104405    D.QQAD= 000000    FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060    DEVIL   005114R 025 D.QUAD= 000000    FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061    DEVNTF= 104470    D.RES = 000000    FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001    DIGITS  000006R 025 D.RHO = 000000    FD.INS= 000010
CHAN12  002676R     025  CLRSI   000000RG  020 DIMAX = 000020    D.ROT = 000003    FD.ISP= 002000
CHAN13  000022R     023  CLRS1   000010RG  020 DIMBLK  003112R 025 D.SCON= 000006    FD.MNT= 100000
CHAN14  000204R     023  CMDBUF  003250R  025 DIMPTR  003114R 025 D.SDIF= 000006    FD.OSP= 004000
CHNOPN= 104471           CMDMAX= 000010        DIMTYP  003116R 025 D.SUB = 000000    FD.PLC= 000004
CHRAND= 000051           CNTSIZ  005130R  025 DIRFUL= 104466       D.SUBS= 000006    FD.PRN= 000004
CHRBEL= 000007           COMBIN  005076R  025 DOMERR= 104417       D.TAKE= 000002    FD.PSE= 010000
CHRBKR= 000133           COMCNT  005126R  025 DPTERR= 104416       D.TRAN= 000004    FD.RAH= 000001
CHRBOL= 000200           COMERR= 104425        DYDCNT  003224R 025 D.UNON= 000006    FD.RAN= 000002
CHRBOX= 000114           COMPTR  005122R  025 D.AND = 000000       D.$MAX= 000000    FD.REC= 000001
CHRCMA= 000054           COMRTN  003156R  025 D.BXCT= 000007       D.$MIN= 000000    FD.RWM= 000001
CHRCOL= 000076           COMSKP  005124R  025 D.BXIO= 000007       D.$MUL= 000000    FD.SDI= 000020
CHRDEL= 000107           CORPTR  005120R  025 D.BXNL= 000007       D.$PLU= 000000    FD.SQD= 000040
CHRDIV= 000053           CS.ASS= 000200        D.BXPP= 000007       D.$RES= 000000    FD.TTY= 000004
CHRDLR= 000176           CS.BOO= 000010        D.BXPW= 000007       D.$SUB= 000000    FD.WBH= 000002
CHRDLT= 000110           CS.FLO= 000020        D.BXRL= 000007       D.1COM= 000001    FF.CHR= 000005
CHRDOT= 000056           CS.IDN= 000040        D.CIRC= 000001       D.1EXP= 000002    FF.NV = 000003
CHREP = 000105           CS.ONE= 000100        D.COMB= 000001       D.1ROT= 000003    FF.POE= 000002
CHRERR= 104431           CTEPS   000152R  025 D.COMP= 000001       EDICHR  005063R 025 FF.RWD= 000001
CHRFIL= 000177           CTMAX   000212R  025 D.CONT= 000006       EDICNT  005064R 025 FF.RWF= 000006
CHRFRO= 000135           CTRMAX  000172R  025 D.CTLM= 000000       EDIFLG  005062R 025 FF.SPC= 000004
```

SYMBOL TABLE

```
FHEADR  005012R     025 F.FLAG= 000002        HSHMSK= 177601        MONBLK  003120R     025 M.RDCT= 000011
FHRET   005040R     025 F.FLB = 000100        HSHSIZ= 000200        MONCVT  003132R     025 M.REVS= 000003
FILNTF= 104467          F.FNAM= 000110        ICHAN   005050R     025 MONLEN  003130R     025 M.ROLL= 000000
FIS   = 000000          F.FNB = 000102        ILFNAM= 104464        MONPTR  003122R     025 M.SATA= 000001
FISCON= 046000          F.FNDF= 000010        ILLCHN= 104540        MONSFT  003126R     025 M.SCAN= 000011
FIXPTR  003074R     025 F.FTYP= 000116        ILLSW = 104565        MONTYP  003124R     025 M.SHAP= 000000
FLAGS   004764R     025 F.FUN = 000010        INDERR= 104410        MONVAL  003134R     025 M.SIGN= 000000
FLCBUF  003250R     025 F.FVER= 000120        INNTMP  003250R     025 M.ABS = 000000        M.SUBR= 000001
FLINE   005014R     025 F.HEAD= 000006        INNTM2  005016R     025 M.BOL = 000200        M.SYM = 000203
FLOEXP  003270R     025 F.HIBK= 000004        INPTR   000064R     025 M.BXAV= 000007        M.SYN = 000201
FLONEG  003272R     025 F.LBL = 000020        INSNAM= 104462            M.BXCR= 000007        M.TRAN= 000012
FLSWTB  000222R     025 F.LINE= 000012        IOERR = 104477            M.BXCT= 000007        M.$ABS= 000000
FLTLEN= 000010          F.LOC = 000004        I.CNT = 177776            M.BXEX= 000007        M.$CEI= 000000
FNEXIT= ****** GX       F.LSTC= 000004        I.DIM = 177772            M.BXFX= 000007        M.$FLR= 000000
FNINSI  005044R     025 F.LUN = 000042        I.LEN = 000012            M.BXIO= 000007        M.$IDN= 000000
FORIGN  000032R     025 F.MBCT= 000054        I.OERR= 000062            M.BXLC= 000003        M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055        I.OLD = 177770            M.BXNC= 000007        M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056        I.PTR = 177774            M.BXNL= 000007        M.1RDT= 000011
FO.RD = 000001          F.MON = 000002        I.STR = 177766            M.BXPP= 000007        M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001        KILL  = ****** GX         M.BXPW= 000007        M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024        LABERR= 104406            M.BXRL= 000007        NAMBUF  003240R     025
FPMODE  000042R     025 F.NREC= 000030        LABLPT  005042R     025 M.BXTS= 000007        NB.DEV= 000200
FPP   = 000000          F.OPN = 000010        LARGBL  003162R     025 M.BXWA= 000007        NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030        LARGCV  003176R     025 M.CEI = 177777        NB.NAM= 000004
FREBLK  007400R     025 F.PREV= 000000        LARGFL  003160R     025 M.CHR = 000202        NB.SD1= 000400
FUZZ    000142R     025 F.PRNT= 100000        LARGLE  003174R     025 M.CMT = 000206        NB.SD2= 001000
FXPTR1  003076R     025 F.PROT= 010000        LARGLN  003166R     025 M.DOUT= 000006        NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016        LARGPT  003164R     025 M.DQAD= 000006        NB.STP= 000020
F$EVI = 000200          F.RATT= 000001        LARGSF  003172R     025 M.ENCD= 000004        NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034        LARGTY  003170R     025 M.EXE = 000000        NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017        LARGVA  003200R     025 M.EXP = 000001        NB.VER= 000001
F.ACTL= 000076          F.READ= 004000        LBINC   005100R     025 M.FACT= 000001        ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040        LBUFF   003316R     025 M.FIL = 000177        ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040        LBUFLN= 000621            M.FLR = 177777        ND.NAM= 000004
F.BDB = 000070          F.RET = 177772        LCOUNT  005106R     025 M.FMT = 000004        ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002        LENERR= 104412            M.GDUP= 000003        ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000        LENTAB  000054R     025 M.GDWN= 000003        ND.$E$= 000020
F.BKDN= 000026          F.R2  = 000014        LEXPTR  003100R     025 M.GETI= 000006        ND.$NM= 000000
F.BKDS= 000020          F.R3  = 000016        LFTOP   005066R     025 M.IBM = 000003        ND.$N$= 000040
F.BKEF= 000050          F.R4  = 000020        LINC    005102R     025 M.IDN = 000000        ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100        LINE    005016R     025 M.INSP= 000006        NEGSGN  005063R     025
F.BKST= 000024          F.SPDV= 000072        LININC  005026R     025 M.IOTA= 000000        NINPT1  003302R     025
F.BKVB= 000064          F.SPUN= 000074        LISIZE= 000020            M.LIT = 000205        NINPT2  003300R     025
F.BR  = 000020          F.STBK= 000036        LITOP   005072R     025 M.LN  = 000001        NINSTK  003304R     025
F.CHR = 000075          F.STK = 000022        LNLNG = 104541            M.MMD = 000010        NINTMP  003310R     025
F.CNTG= 000034          F.STOP= 000004        LOCLCT  005036R     025 M.NEG = 000000        NINTYP  003306R     025
F.CODE= 000010          F.STP = 000001        LOWCOR  000002R     025 M.NOT = 000000        NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002        LSTERP  004766R     025 M.NUB = 000006        NOTOPN= 104473
F.DSPT= 000044          F.TRC = 000002        LWIDTH  000070R     025 M.NUM = 000204        NPRT1   004762R     025
F.DVNM= 000134          F.UNIT= 000136        L.BITS= 000003            M.OUTR= 000012        NULL    007364R     025
F.EFBK= 000010          F.URBD= 000020        L.CNT = 000002            M.PITM= 000000        NUMCHN= 000020
F.EFN = 000050          F.VAL = 177776        L.PRNT= 000001            M.PRDS= 000006        NUMDFL  003230R     025
F.END = 002000          F.VBN = 000064        L.PTR = 000000            M.QQAD= 000000        NUMEFL  003233R     025
F.EOBB= 000032          F.VBSZ= 000060        L.PTR1= 000004            M.QUAD= 000000        NUMESN  003236R     025
F.ERR = 000052          GNOME   003102R     025 L.SUBS= 000002            M.RAND= 000012        NUMFLG  003230R     025
F.FACC= 000043          HASHC   005136R     025 MACHIN= 000106            M.RAVL= 000000        NUMLEN  000062R     025
F.FFBY= 000014          HASHTB  005140R     025 MAXCHN= 000014            M.RCIP= 000000        NUMNEG  003232R     025
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 46-3
SYMBOL TABLE

NUMQFL  003234R     025 RANDAC  003064R     025 R.LNUM= 000010          SY.BRK= 100000          T.LIST= 000400
NUMRIG  003231R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BS = 000010          T.LIT = 000020
NUMWID  003226R     025 RANERR= 104411          R.OFF = 000000          SY.CLR= 000040          T.MON = 000010
NUMZFL  003235R     025 RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL = 000004
NYIERR= 104432          RARGCV  003220R     025 R.PTR = 000000          SY.NOU= 000100          T.NUM = 002000
N.BLSZ= 000036          RARGFL  003202R     025 R.SEQ = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLE  003216R     025 R.VAR = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGLN  003210R     025 R6    =%000006          SY.SRT= 000400          T.SOP = 000002
N.DEVN= 000034          RARGPT  003206R     025 R7    =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGSF  003214R     025 SI    =%000005          SY.SVL= 000200          T.TMP = 000200
N.DVNM= 000032          RARGTY  003212R     025 SIDAMN= 104433          S.AB  = 177776          T.VAL = 100000
N.EXT = 000014          RARGVA  003222R     025 SIFRAM  003314R     025 S.ASB = 100000          T.VAR = 001000
N.FID = 000000          RBASE   005074R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER = 177400
N.FNAM= 000006          RCOUNR  005112R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          RCOUNT  005110R     025 SITP    007144R     025 S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDCNT  005070R     025 SIWARN= 000200          S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDIGN  005066R     025 SIZTAB  000044R     025 S.FDB = 000140          VARBLS  000000R     025
N.NEXT= 000022          REDOFF  005072R     025 SI.BIT= 177774          S.FLB = 040000          VERNUM  000000R     025
N.PPN = 000024          REENTR  003060R     025 SI.LEN= 177772          S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESBIT  003142R     025 SI.RET= 177772          S.FNB = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.VAL= 177776          S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESERR= 104440          SKPSIZ  005134R     025 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESSFT  003146R     025 STKINC= 000774          S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESTYP  003144R     025 STOP  = 104420          S.NFEN= 000020          $FIS  = ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FPP  = ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL = 000000          $FPPLN= ****** G
OPTMP4  005076R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $RSTS = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSX  = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RT11 = ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          .BOOLE= 000002
OPTMP9  005110R     025 RSTS  = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .CHARA= 000000
OPTM10  005112R     025 RSX   = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .DEF. = 000002
OPTM14  005026R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436          .FLOAT= 000006
ORIGIN  000030R     025 RT11  = 000000          SYMWRN= 000054          TYPE  = 000166          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003          SYNERR= 104407          T.BOL = 020000          .INTEG= 000004
OVROPR= 000000          R.FIX = 000001          SYSERR= 104402          T.CON = 004000          .MOLUN  000404R     027
PATCH   007344R     025 R.LCT = 000002          SYSTAT  000000R     024 T.DIM = 000100          ...GBL= 000000
PKILL = ****** GX       R.LN  = 000002          SY.BAD= 000001          T.FUN = 010000          ...TPC= 000020
PRECSN  000052R     025

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

```
CLRSI   MACRO M1113  08-FEB-80 11:06  PAGE 46-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000220      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12943 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:06
CLRSI,DM1:[305,301]CLRSI/-SP=APLRX1,APLDEF,CLRSI,END
```

```
**FILE**ID**CLSFIL
```

```
  CCCCCCCC   LL            SSSSSSSS   FFFFFFFFFF     IIIIII     LL
  CCCCCCCC   LL            SSSSSSSS   FFFFFFFFFF     IIIIII     LL
CC           LL          SS           FF               II       LL
CC           LL          SS           FF               II       LL
CC           LL          SS           FF               II       LL
CC           LL          SS           FF               II       LL
CC           LL            SSSSSS     FFFFFFFF         II       LL
CC           LL            SSSSSS     FFFFFFFF         II       LL
CC           LL                  SS   FF               II       LL
CC           LL                  SS   FF               II       LL
CC           LL                  SS   FF               II       LL                ....
CC           LL                  SS   FF               II       LL                ....
  CCCCCCCC   LLLLLLLLLL   SSSSSSSS     FF             IIIIII     LLLLLLLLLL        ....
  CCCCCCCC   LLLLLLLLLL   SSSSSSSS     FF             IIIIII     LLLLLLLLLL        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:08
TABLE OF CONTENTS

2- 1 APLDEF
2- 8 A P L
2- 9
2- 10 A PROGRAMMING LANGUAGE
2- 11
2- 12 STEVE POULSEN & DAVE VANN
2- 13 11-Jul-76
2- 14 ROBERT GOTTLIEB & PETER HAYNES
2- 15 2-Feb-79
2- 16
2- 17 V2
2- 18
2- 19
2- 20 COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
2- 21 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
2- 22
2- 23
2- 24 THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
2- 25 COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
2- 26 COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
2- 27 BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
2- 28 USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
2- 29 TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
2- 30 DEC.
2- 31
2- 32 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
2- 33 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
2- 34 CORPORATION.
2- 35
2- 36 DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
2- 37 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
2- 38
2- 39
3- 46 DEFINITIONS ASSEMBLY PARAMETERS
4- 139 DEFINITIONS OPERATOR OVERLAYS
5- 268 DEFINITIONS REGISTERS
6- 331 DEFINITIONS MEMORY ORGANIZATION
7- 377 DEFINITIONS ORDERING PSECTS
8- 426 DEFINITIONS SI PARSE FRAMES
9- 474 DEFINITIONS TOKEN DESCRIPTION BITS
10- 515 DEFINITIONS SI EXECUTION FRAMES
11- 580 DEFINITIONS SPECIAL PARSE CHARACTERS
12- 607 DEFINITIONS CHRTAB BITS
12- 624 DEFINITIONS SYSTEM STATUS BITS
13- 640 DEFINITIONS F-FRAMES
14- 724 DEFINITIONS FUNCTION LINE BLOCKS
16- 807 DEFINITIONS ARRAY STORAGE BLOCK
17- 869 DEFINITIONS LIST STORAGE BLOCK
18- 911 DEFINITIONS INDEXING STACK FRAME
19- 940 DEFINITIONS SYMBOL TABLE ENTRY
20- 969 DEFINITIONS CHANNEL BLOCK
22- 1085 DEFINITIONS RSX DATA
23- 1110 DEFINITIONS NAME BLOCKS
33- 1612 DEFINITIONS MACROS
35- 1905 DEFINITIONS APL CHARACTERS
36- 1967 DEFINITIONS ERRORS
44- 4 COMMANDS CLOSE

CLSFIL MACRO M1113 08-FEB-80 11:08 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\

```

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
CLSFIL  MACRO M1113  08-FEB-80 11:08  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  CLSFIL
                                .IDENT  \V02.03\
                                
                                .SBTTL  COMMANDS                CLOSE
                                
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       CLOSE A SPECIFIED CHANNEL
                                ;--
000000                          .PSECT  CODE
                                
000000                          CLSFIL::
000000                          CALL    CMDCHN          ;GET A CHANNEL NUMBER
000004                          CALLR   RESCHN          ;RESET CHANNEL
```

```
CLSFIL  MACRO M1113  08-FEB-80 11:08  PAGE 46
COMMANDS                CLOSE

      1          000001                              .END
```

```
CLSFIL  MACRO M1113  08-FEB-80 11:08  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.ALOC= 000162    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLPR= 000072    C.BLOK= 000064    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRMNS= 000137    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMUL= 000075    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRNEG= 000100    C.BUFL= 177762    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRPDL= 000271    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPLU= 000055    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPWR= 000120    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRQST= 000121    C.DIG = 000040    D.EXPD= 000002    ERRO    004774R     025
A.ALEN= 000004           CHRQUO= 000113    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRRBR= 000047    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R     025
A.ASB = 000200           CHRRPR= 000042    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.BITS= 000003           CHRSEM= 000074    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.CON = 000020           CHRSMO= 000112    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSPA= 000040    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIM1= 000006           CHRUPA= 000131    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.NUM = 000006           CHR.BS= 000010    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.PTR = 000000           CHR.CC= 000003    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.TMP = 000010           CHR.CR= 000015    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
BADDIR= 104463           CHR.CZ= 000032    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADERR= 104437           CHR.GT= 000046    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BLZBUB  005116R     025  CHR.LF= 000012    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LT= 000043    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
CCNT    005072R     025  CHR.SI= 000017    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN04  001056R     025  CHR..E= 000145    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN05  001240R     025  CHR..S= 000163    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN06  001422R     025  CHR..T= 000164    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN07  001604R     025  CHR..U= 000165    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN08  001766R     025  CHR..Z= 000172    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN09  002150R     025  CHR..0= 000060    DEVIL   005114R 025 D.QUAD= 000000    FD.F11= 040000
CHAN10  002332R     025  CHR..1= 000061    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN11  002514R     025  CH.AND= 000001    DIGITS  000006R 025 D.RHO = 000000    FD.ISP= 002000
CHAN12  002676R     025  CLSFIL  000000RG  020 DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN13  000022R     023  CMDBUF  003250R  025 DIMBLK  003112R 025 D.SCON= 000006    FD.OSP= 004000
CHAN14  000204R     023  CMDCHN= ****** GX     DIMPTR  003114R 025 D.SDIF= 000006    FD.PLC= 000004
CHNOPN= 104471           CMDMAX= 000010        DIMTYP  003116R 025 D.SUB = 000000    FD.PRN= 000004
CHRAND= 000051           CNTSIZ  005130R  025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRBEL= 000007           COMBIN  005076R  025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBKR= 000133           COMCNT  005126R  025 DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBOL= 000200           COMERR= 104425        DYDCNT  003224R 025 D.UNON= 000006    FD.REC= 000001
CHRBOX= 000114           COMPTR  005122R  025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRCMA= 000054           COMRTN  003156R  025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCOL= 000076           COMSKP  005124R  025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRDEL= 000107           CORPTR  005120R  025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDIV= 000053           CS.ASS= 000200        D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDLR= 000176           CS.BOO= 000010        D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLT= 000110           CS.FLO= 000020        D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDOT= 000056           CS.IDN= 000040        D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHREP = 000105           CS.ONE= 000100        D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRERR= 104431           CTEPS   000152R  025 D.COMP= 000001    EDICHR  005063R 025 FF.RWF= 000006
CHRFIL= 000177           CTMAX   000212R  025 D.CONT= 000006    EDICNT  005064R 025 FF.SPC= 000004
CHRFRO= 000135           CTRMAX  000172R  025 D.CTLM= 000000    EDIFLG  005062R 025 FHEADR  005012R     025
```

CLSFIL MACRO M1113 08-FEB-80 11:08 PAGE 46-2
SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110        ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102        ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010        ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116        ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FIXPTR  003074R     025 F.FUN = 000010        INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R     025 F.FVER= 000120        INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R     025 F.HEAD= 000006        INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
FLINE   005014R     025 F.HIBK= 000004        INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R     025 F.LBL = 000020        INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R     025 F.LINE= 000012        IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R     025 F.LOC = 000004        I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004        I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R     025 F.LUN = 000042        I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R     025 F.MBCT= 000054        I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055        I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056        I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002        I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001        LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FO.WRT= 000016          F.NRBD= 000024        LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R     025 F.NREC= 000030        LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FPP   = 000000          F.OPN = 000010        LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030        LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R     025 F.PREV= 000000        LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R     025 F.PRNT= 100000        LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R     025 F.PROT= 010000        LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016        LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001        LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034        LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017        LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.RD  = 004000        LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040        LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040        LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772        LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002        LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000        LEXFTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014        LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016        LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020        LINE    005016R     025 M.I SP= 000006          NEGSGN  005063R     025
F.BKP1= 000051          F.SEQN= 000100        LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.BKST= 000024          F.SPDV= 000072        LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.BKVB= 000064          F.SPUN= 000074        LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.BR  = 000020          F.STBK= 000036        LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CHR = 000075          F.STK = 000022        LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.CNTG= 000034          F.STOP= 000004        LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001        LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002        LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.ERR = 000052          GNOME   003102R     025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FACC= 000043          HASHC   005136R     025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
F.FFBY= 000014          HASHTB  005140R     025 MONBLK  003120R     025 M.RDCT= 000011          NUMQFL  003234R     025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R     025 M.REVS= 000003          NUMRIG  003231R     025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000          NUMWID  003226R     025
```

```
CLSFIL  MACRO M1113  08-FEB-80 11:08  PAGE 46-3
SYMBOL TABLE

NUMZFL  003235R     025 RARGBL  003204R     025 R.OFF  = 000000          SY.CLR= 000040          T.LIT  = 000020
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.ERP= 000020          T.MON  = 000010
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR  = 000000          SY.NOU= 000100          T.NIL  = 000004
N.DESC= 000020          RARGLE  003216R     025 R.SEQ  = 000003          SY.OPN= 000002          T.NUM  = 002000
N.DEV  = 000032         RARGLN  003210R     025 R.VAR  = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034          RARGPT  003206R     025 R6     =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DID  = 000024         RARGSF  003214R     025 R7     =%000007          SY.SVC= 000004          T.SOP  = 000002
N.DVNM= 000032          RARGTY  003212R     025 SI     =%000005          SY.SVL= 000200          T.SPOP= 000040
N.EXT  = 000014         RARGVA  003222R     025 SIDAMN= 104433           S.AB   = 177776          T.TMP  = 000200
N.FID  = 000000         RBASE   005074R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAL  = 100000
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R     025 S.BFHD= 000020          T.VAR  = 001000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R     025 S.BITS= 000002          UPPER  = 177400
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R     025 S.CHR1= 000003          UTNAME  000344R     023
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLE  007400R     025
N.PPN  = 000024         REENTR  003060R     025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R     025
N.UNIT= 000034          RESCHN= ****** GX       SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
OBUFLN= 001000          RESCVT  003152R     025 SI.VAL= 177776           S.FNBW= 000017          WIDTH   000072R     025
OCHAN   005052R     025 RESERR= 104440           SKPSIZ  005134R     025 S.FNTY= 000004          WROUND  000020R     025
OMODE   000402R     023 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP0  005066R     025 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSFULL= 104403
OPTMP1  005070R     025 RESSFT  003146R     025 STKINC= 000774           S.LINK= 177774          WSNAME  000074R     025
OPTMP2  005072R     025 RESTYP  003144R     025 STOP   = 104420          S.NFEN= 000020          WSTIME  000132R     025
OPTMP3  005074R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP4  005076R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP5  005100R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP7  005104R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP8  005106R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSX   = ****** G
OPTMP9  005110R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          $RT11  = ****** G
OPTM10  005112R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .BOOLE= 000002
OPTM14  005026R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .CHARA= 000000
ORIGIN  000030R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436          .DEF.  = 000002
OUTPTR  000066R     025 RT11   = 000000          SYMWRN= 000054          TYPE   = 000166          .FLOAT= 000006
OVROPR= 000000              R.BITS= 000003       SYNERR= 104407          T.BOL  = 020000          .FSRCB= ****** G
PATCH   007344R     025 R.FIX  = 000001          SYSERR= 104402          T.CON  = 004000          .INTEG= 000004
PRECSN  000052R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100          .MOLUN  000404R     023
RANDAC  003064R     025 R.LN   = 000002          SY.BAD= 000001          T.FUN  = 010000          ...GBL= 000000
RANDY   003062R     025 R.LNUM= 000010           SY.BRK= 100000          T.LIST= 000400           ...TPC= 000020
RANERR= 104411              R.NEXT= 000006       SY.BS  = 000010

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

```
CLSFIL  MACRO M1113  08-FEB-80 11:08  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000010      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12904 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:02
CLSFIL,DM1:[305,301]CLSFIL/-SP=APLRX1,APLDEF,CLSFIL,END
```

FILEID**CMDCHN

```
  CCCCCCCC    MM      MM    DDDDDDDD        CCCCCCCC    HH      HH    NN      NN
  CCCCCCCC    MM      MM    DDDDDDDD        CCCCCCCC    HH      HH    NN      NN
CC            MMMM  MMMM    DD      DD    CC          HH      HH    NN      NN
CC            MMMM  MMMM    DD      DD    CC          HH      HH    NN      NN
CC            MM  MM  MM    DD      DD    CC          HH      HH    NNNN    NN
CC            MM  MM  MM    DD      DD    CC          HH      HH    NNNN    NN
CC            MM      MM    DD      DD    CC          HHHHHHHHHH    NN  NN  NN
CC            MM      MM    DD      DD    CC          HHHHHHHHHH    NN  NN  NN
CC            MM      MM    DD      DD    CC          HH      HH    NN    NNNN
CC            MM      MM    DD      DD    CC          HH      HH    NN    NNNN
CC            MM      MM    DD      DD    CC          HH      HH    NN      NN      ....
  CCCCCCCC    MM      MM    DDDDDDDD        CCCCCCCC    HH      HH    NN      NN      ....
  CCCCCCCC    MM      MM    DDDDDDDD        CCCCCCCC    HH      HH    NN      NN      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:10
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CMDCHN           GET CHANNEL NUMBER FOR COMMAND
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     =  'M         ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:10  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CMDCHN
                                .IDENT  \V02.03\

                                .SBTTL  CMDCHN          GET CHANNEL NUMBER FOR COMMAND

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS ROUTINE IS PASSED A STRING CONTAINING A NUMBER.
                                ;       THE NUMBER IS EXTRACTED, THE VALUE CHECKED, AND A POINTER
                                ;       TO THE APPROPRIATE CHANNEL BLOCK RETURNED
                                ;
                                ; INPUTS:
                                ;       R2 -    CHR
                                ;       R3 -    LENGTH OF TEXT
                                ;       R4 -    POINTER TO TEXT
                                ;
                                ; OUTPUTS:
                                ;       R2 -    POINTER TO CHANNEL BLOCK
                                ;       R3 -    LENGTH OF REMAINING TEXT
                                ;       R4 -    POINTER TO REMAINING TEXT
                                ;--
000000                                  .PSECT  CODE

000000                          CMDCHN::
000000  120227  000040                  CMPB    R2,#CHRSPA      ;SKIP LEADING BLANKS
000004  001004                          BNE     1$              ;NO OTHER CHRS
000006                                  GETC    2$              ;GET ANOTHER CHR
000006  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000010  002425                          BLT 2$                  ;NO, BRANCH TO 2$
000012  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000014  000771                          BR      CMDCHN          ;CHECK IT OUT

000016                          1$:     CALL    NUMBER          ;GET A NUMBER
000022  005203                          INC     R3              ;BACK UP COUNTER
000024  005304                          DEC     R4              ;BACK UP POINTER
000026                                  CALL    F.TO.I          ;Convert to Integer
000032  000414                          BR      2$              ;Illegal Channel Number
000034  013702  003154'                 MOV     RESVAL,R2       ;Get the Channel Number
000040  003411                          BLE     2$              ;Must be Positive
000042  020227  000014                  CMP     R2,#MAXCHN      ;Within Range?
000046  101006                          BHI     2$              ;No
000050  006302                          ASL     R2              ;Multiply Channel # by 2
000052  016202  000000G                 MOV     CHANNEL(R2),R2  ;Get Pointer
000056  121427  000040                  CMPB    (R4),#CHRSPA    ;FOLLOWED BY SPACE?
000062                                  RETURN                  ;WE'LL NEVER KNOW!!

000064  104540                  2$:     ILLCHN                  ;ILLEGAL CHANNEL NUMBER
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 46
CMDCHN           GET CHANNEL NUMBER FOR COMMAND

      1          000001                          .END
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135    C.ALOC= 000162    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLBR= 000073    C.BLOK= 000064    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRLPR= 000072    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMNS= 000137    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRMUL= 000075    C.BUFL= 177762    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRNEG= 000100    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPDL= 000271    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPLU= 000055    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRPWR= 000120    C.DIG = 000040    D.EXPD= 000002    ERR0    004774R     025
A.ALEN= 000004           CHRQST= 000121    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRQUO= 000113    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R     025
A.ASB = 000200           CHRRBR= 000047    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.BITS= 000003           CHRRPR= 000042    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.CON = 000020           CHRSEM= 000074    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSMO= 000112    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIM1= 000006           CHRSPA= 000040    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.NUM = 000006           CHRUPA= 000131    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.PTR = 000000           CHR.BS= 000010    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.TMP = 000010           CHR.CC= 000003    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
BADDIR= 104463           CHR.CR= 000015    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADERR= 104437           CHR.CZ= 000032    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BLZBUB  005116R     025  CHR.GT= 000046    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LF= 000012    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
CCNT    005072R     025  CHR.LT= 000043    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CHANEX  003060R     025  CHR.SI= 000017    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHANNE= ****** GX        CHR.SL= 000057    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVIL   005114R 025 D.QUAD= 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R 025 D.RHO = 000000    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R 025 DIMBLK  003112R 025 D.SCON= 000006    FD.OSP= 004000
CHAN13  000022R     023  CMDCHN  000000RG 020 DIMPTR  003114R 025 D.SDIF= 000006    FD.PLC= 000004
CHAN14  000204R     023  CMDMAX= 000010    DIMTYP  003116R 025 D.SUB = 000000    FD.PRN= 000004
CHNOPN= 104471           CNTSIZ  005130R 025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051           COMBIN  005076R 025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007           COMCNT  005126R 025 DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133           COMERR= 104425    DYDCNT  003224R 025 D.UNON= 000006    FD.REC= 000001
CHRBOL= 000200           COMPTR  005122R 025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114           COMRTN  003156R 025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054           COMSKP  005124R 025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076           CORPTR  005120R 025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.BOO= 000010    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176           CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110           CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056           CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105           CTEPS   000152R 025 D.COMP= 000001    EDICHR  005063R 025 FF.RWF= 000006
CHRERR= 104431           CTMAX   000212R 025 D.CONT= 000006    EDICNT  005064R 025 FF.SPC= 000004
CHRFIL= 000177           CTRMAX  000172R 025 D.CTLM= 000000    EDIFLG  005062R 025 FHEADR  005012R     025
```

SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110      H.HSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000
FILNTF= 104467          F.FNB = 000102      ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
FIS   = 000000          F.FNDF= 000010      ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116      ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FIXPTR  003074R     025 F.FUN = 000010      ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FLAGS   004764R     025 F.FVER= 000120      INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLCBUF  003250R     025 F.HEAD= 000006      INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
FLINE   005014R     025 F.HIBK= 000004      INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
FLOEXP  003270R     025 F.LBL = 000020      INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
FLONEG  003272R     025 F.LINE= 000012      INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLSWTB  000222R     025 F.LOC = 000004      IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 000004      I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FNINSI  005044R     025 F.LUN = 000042      I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055      I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056      I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.RD = 000001          F.MON = 000002      I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001      I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FPMODE  000042R     025 F.NREC= 000030      LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FPP   = 000000          F.OPN = 000010      LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030      LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FREBLK  007400R     025 F.PREV= 000000      LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
FUZZ    000142R     025 F.PRNT= 100000      LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
FXPTR1  003076R     025 F.PROT= 010000      LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016      LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$EVI = 000200          F.RATT= 000001      LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034      LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017      LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F.ACTL= 000076          F.READ= 004000      LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040      LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040      LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BDB = 000070          F.RET = 177772      LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002      LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000      LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
F.BKDN= 000026          F.R2  = 000014      LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKDS= 000020          F.R3  = 000016      LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
F.BKEF= 000050          F.R4  = 000020      LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100      LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BKST= 000024          F.SPDV= 000072      LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.BKVB= 000064          F.SPUN= 000074      LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.BR  = 000020          F.STBK= 000036      LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.CHR = 000075          F.STK = 000022      LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CNTG= 000034          F.STOP= 000004      LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.CODF= 000010          F.STP = 000001      LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002      LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.DSPT= 000044          F.TO.I= ****** GX   LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.DV.M= 000134          F.TRC = 000002      L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.EFBK= 000010          F.UNIT= 000136      L.CNT = 000002          M.PITM= 000000          NUMBER= ****** GX
F.EFN = 000050          F.URBD= 000020      L.PRNT= 000001          M.PRDS= 000006          NUMCHN= 000020
F.END = 002000          F.VAL = 177776      L.PTR = 000000          M.QQAD= 000000          NUMDFL  003230R     025
F.EOBB= 000032          F.VBN = 000064      L.PTR1= 000004          M.QUAD= 000000          NUMEFL  003233R     025
F.ERR = 000052          F.VBSZ= 000060      L.SUBS= 000002          M.RAND= 000012          NUMESN  003236R     025
F.FACC= 000043          GNOME   003102R     025 MACHIN= 000106      M.RAVL= 000000          NUMFLG  003230R     025
F.FFBY= 000014          HASHC   005136R     025 MAXCHN= 000014      M.RCIP= 000000          NUMLEN  000062R     025
F.FLAG= 000002          HASHTB  005140R     025 MONBLK  003120R     025 M.RDCT= 000011          NUMNEG  003232R     025
F.FLB = 000100          HSHMSK= 177601          MONCVT  003132R     025 M.REVS= 000003          NUMQFL  003234R     025
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 46-3
SYMBOL TABLE

NUMRIG  003231R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BS  = 000010          T.LIT  = 000020
NUMWID  003226R     025 RANERR= 104411          R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
NUMZFL  003235R     025 RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
NYIERR= 104432          RARGCV  003220R     025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
N.BLSZ= 000036          RARGFL  003202R     025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLE  003216R     025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGLN  003210R     025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DEVN= 000034          RARGPT  003206R     025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGSF  003214R     025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.DVNM= 000032          RARGTY  003212R     025 SIDAMN= 104433          S.AB   = 177776          T.VAL  = 100000
N.EXT = 000014          RARGVA  003222R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAR  = 001000
N.FID = 000000          RBASE   005074R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER  = 177400
N.FNAM= 000006          RCOUNR  005112R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          RCOUNT  005110R     025 SITP    007144R     025 S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDCNT  005070R     025 SIWARN= 000200          S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDIGN  005066R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLS  000000R     025
N.NEXT= 000022          REDOFF  005072R     025 SI.BIT= 177774          S.FLB  = 040000          VERNUM  000000R     025
N.PPN  = 000024         REENTR  003060R     025 SI.LEN= 177772          S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESBIT  003142R     025 SI.RET= 177772          S.FNB  = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.VAL= 177776          S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESERR= 104440          SKPSIZ  005134R     025 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESSFT  003146R     025 STKINC= 000774          S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESTYP  003144R     025 STOP   = 104420          S.NFEN= 000020          $FIS   = ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FPP   = ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPPLN= ****** G
OPTMP4  005076R     025 R:NC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RT11  = ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          .BOOLE= 000002
OPTMP9  005110R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .CHARA= 000000
OPTM10  005112R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .DEF.  = 000002
OPTM14  005026R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436          .FLOAT= 000006
ORIGIN  000030R     025 RT11   = 000000          SYMWRN= 000054          TYPE   = 000166          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003          SYNERR= 104407          T.BOL  = 020000          .INTEG= 000004
OVROPR= 000000          R.FIX  = 000001          SYSERR= 104402          T.CON  = 004000          .MOLUN  000404R     023
PATCH   007344R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100          ...GBL= 000000
PRECSN  000052R     025 R.LN   = 000002          SY.BAD= 000001          T.FUN  = 010000          ...TPC= 000020
RANDAC  003064R     025 R.LNUM= 000010          SY.BRK= 100000          T.LIST= 000400

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
CMDCHN  MACRO M1113  08-FEB-80 11:10  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000066      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12914 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:00
CMDCHN,DM1:[305,301]CMDCHN/-SP=APLRX1,APLDEF,CMDCHN,END
```

FILEID**CMDNBR

```
  CCCCCCCC    MM       MM    DDDDDDDD      NN       NN    BBBBBBBB      RRRRRRRR
  CCCCCCCC    MM       MM    DDDDDDDD      NN       NN    BBBBBBBB      RRRRRRRR
CC            MMMM   MMMM    DD       DD   NN       NN    BB      BB    RR      RR
CC            MMMM   MMMM    DD       DD   NN       NN    BB      BB    RR      RR
CC            MM  MM  MM    DD       DD    NNNN     NN    BB      BB    RR      RR
CC            MM  MM  MM    DD       DD    NNNN     NN    BB      BB    RR      RR
CC            MM      MM    DD       DD    NN  NN   NN    BBBBBBBB      RRRRRRRR
CC            MM      MM    DD       DD    NN  NN   NN    BBBBBBBB      RRRRRRRR
CC            MM      MM    DD       DD    NN    NNNN    BB      BB    RR  RR
CC            MM      MM    DD       DD    NN    NNNN    BB      BB    RR  RR
CC            MM      MM    DD       DD    NN      NN    BB      BB    RR    RR        ....
CC            MM      MM    DD       DD    NN      NN    BB      BB    RR    RR        ....
  CCCCCCCC    MM      MM    DDDDDDDD      NN      NN    BBBBBBBB      RR      RR      ....
  CCCCCCCC    MM      MM    DDDDDDDD      NN      NN    BBBBBBBB      RR      RR      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:12
TABLE OF CONTENTS


```
     2-    1     APLDEF
     2-    8     A P L
     2-    9
     2-   10     A PROGRAMMING LANGUAGE
     2-   11
     2-   12     STEVE POULSEN & DAVE VANN
     2-   13     11-Jul-76
     2-   14     ROBERT GOTTLIEB & PETER HAYNES
     2-   15     2-Feb-79
     2-   16
     2-   17     V2
     2-   18
     2-   19
     2-   20     COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25     COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26     COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27     BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28     USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29     TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30     DEC.
     2-   31
     2-   32     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33     AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34     CORPORATION.
     2-   35
     2-   36     DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37     SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46     DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139     DEFINITIONS      OPERATOR OVERLAYS
     5-  268     DEFINITIONS      REGISTERS
     6-  331     DEFINITIONS      MEMORY ORGANIZATION
     7-  377     DEFINITIONS      ORDERING PSECTS
     8-  426     DEFINITIONS      SI PARSE FRAMES
     9-  474     DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515     DEFINITIONS      SI EXECUTION FRAMES
    11-  580     DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607     DEFINITIONS      CHRTAB BITS
    12-  624     DEFINITIONS      SYSTEM STATUS BITS
    13-  640     DEFINITIONS      F-FRAMES
    14-  724     DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807     DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869     DEFINITIONS      LIST STORAGE BLOCK
    18-  911     DEFINITIONS      INDEXING STACK FRAME
    19-  940     DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969     DEFINITIONS      CHANNEL BLOCK
    22- 1085     DEFINITIONS              RSX DATA
    23- 1110     DEFINITIONS      NAME BLOCKS
    33- 1612     DEFINITIONS      MACROS
    35- 1905     DEFINITIONS      APL CHARACTERS
    36- 1967     DEFINITIONS      ERRORS
    45-   35     COMMANDS                 FUZZ
```

CMDNBR MACRO M1113 08-FEB-80 11:12
TABLE OF CONTENTS

45-	46	COMMANDS	ORIGIN
45-	57	COMMANDS	WIDTH
45-	68	COMMANDS	DIGITS

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 1


      1        000000                  RT11    =  0            ;NOT RT-11
      2        000000                  RSTS    =  0            ;NOT RSTS/E
      3        000115                  RSX     = 'M            ;RSX-11/M
      4
      5        000000                  EIS     =  0            ;NO EIS
      6        000000                  FIS     =  0            ;NO FIS
      7        000000                  FPP     =  0            ;NO FPP
      8
      9        000004                  FPPLEN  =  4.           ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:12  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
      1                                 .TITLE  CMDNBR
      2                                 .IDENT  \V02.03\
      3
      4                                 ;++
      5                                 ; FUNCTIONAL DESCRIPTION:
      6                                 ;       'CMDNBR' FETCHS THE OLD VALUE, ASSIGNS THE NEW VALUE (IF ANY),
      7                                 ;       AND PRINTS THE OLD VALUE
      8                                 ;--
      9 000000                                  .PSECT  CODE
     10
     11 000000                          CMDNBR::
     12 000000                                  PUSH    <R4,R3,R2>      ;SAVE THE REGISTERS
     13 000006  016604  000006                  MOV     6(SP),R4        ;GET THE OP NUMBER
     14 000012                                  CALL    @MONTAB(R4)     ;CALL THE ROUTINE
     15 000016                                  POP     <R2,R3,R4>      ;RESTORE THE REGISTERS
     16 000024                                  CALL    NGET            ;GET INPUT VALUE
     17 000030  000420                          BR      1$              ;NO VALUE, IS QUERY
     18 000032                                  PUSH    <R4,R3,R2>      ;WHAT, AGAIN?
     19 000040  016604  000006                  MOV     6(SP),R4        ;GET THE OP NUMBER AGAIN
     20 000044                                  CALL    @DYDTAB(R4)     ;CALL THE STORE ROUTINE
     21 000050                                  CALL    KILL            ;KILL THE NEW VALUE
     22 000054                                  POP     <R2,R3,R4>      ;RESTORE THE REGISTERS
     23 000062                                  QPRINT  WASL            ;'WAS ''
        000062  012701  000000'                 MOV #WASL,R1            ;POINT TO WASL
     24 000072  005726                  1$:     TST     (SP)+           ;DELETE OP NUMBER
     25 000074  016500  177776                  MOV     SI.VAL(SI),R0   ;POINT TO ARRAY
     26 000100                                  CALL    APRINT          ;PRINT IT
     27 000104                                  CALL    NEWLIN          ;End the line
     28 000110                                  CALLR   KILL            ;WIPE IT OUT
     29
     30 000000                                  .PSECT  TABLES
     31
     32 000000      167     141     163 WASL::  .ASCIZ  /was /
        000003      040     000
     33                                         .EVEN
```

```
                                .SBTTL  COMMANDS                FUZZ
                                ;
                                ;FUZZY -- CHANGE FUZZ FACTOR
                                ;
000114                          .PSECT  CODE

000114                          FUZZY::
000114                                  PUSH    #OP..CT         ;Comparison Tolerance Routine
000120  000727                          BR      CMDNBR          ;Process Command


                                .SBTTL  COMMANDS                ORIGIN
                                ;
                                ;ORIGNN -- CHANGE INDEXING ORIGIN
                                ;
000122                          .PSECT  CODE

000122                          ORIGNN::
000122                                  PUSH    #OP..IO         ;Index Origin
000126  000724                          BR      CMDNBR          ;Process Command


                                .SBTTL  COMMANDS                WIDTH
                                ;
                                ;WIDTTH -- CHANGE LINE WIDTH
                                ;
000130                          .PSECT  CODE

000130                          WIDTTH::
000130                                  PUSH    #OP..PW         ;Print Width
000134  000721                          BR      CMDNBR          ;Process Command


                                .SBTTL  COMMANDS                DIGITS
                                ;
                                ;'DIGIT' -- CHANGE # OF PRINTED DIGITS
                                ;
000136                          .PSECT  CODE

000136                          DIGIT::
000136                                  PUSH    #OP..PP         ;Print Precision
000142  000716                          BR      CMDNBR          ;Process Command
```

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 47
COMMANDS                DIGITS

      1          000001                              .END
```

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 47-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.ALOC= 000162  D.CONT= 000006  EDICNT  005064R  025
AC1    =%000001          CHRLBR= 000073  C.BLOK= 000064  D.CTLM= 000000  EDIFLG  005062R  025
AC2    =%000002          CHRLPR= 000072  C.BLSZ= 000140  D.DEAL= 000003  EDIT   = 000003
AC3    =%000003          CHRMNS= 000137  C.BUFF= 177760  D.DECD= 000002  EIS    = 000000
AC4    =%000004          CHRMUL= 000075  C.BUFL= 177762  D.DIV = 000000  ELESIZ  005132R  025
AC5    =%000005          CHRNEG= 000100  C.CCNT= 177766  D.DMD = 000010  ENDCOR  000004R  025
ALLINT  003274R     025  CHRPDL= 000271  C.CPTR= 177764  D.DQAD= 177777  EOF    = 104475
APRINT= ****** GX        CHRPLU= 000055  C.CZ   = 001000  D.DROP= 000002  ERRMAX= 000166
ARYRET  003276R     025  CHRPWR= 000120  C.DIG  = 000040  D.ENCD= 000002  ERRNUM  004772R  025
ASCQUO= 000042           CHRQST= 000121  C.FASW= 020000  D.EQ   = 000000  ERRPTR  004770R  025
A.ALEN= 000004           CHRQUO= 000113  C.FLEN= 000010  D.EXPD= 000002  ERRO    004774R  025
A.ARG  = 000040          CHRRBR= 000047  C.FLSH= 004000  D.FGET= 000005  EXFBUF  005054R  025
A.ASB  = 000200          CHRRPR= 000042  C.FRC  = 000001  D.FMT  = 000004  E$CT    005056R  025
A.BITS= 000003           CHRSEM= 000074  C.IDEN= 000010  D.FPNT= 000005  E$PT    005060R  025
A.CON  = 000020          CHRSMO= 000112  C.NAME= 000102  D.FPUT= 000005  FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040  C.NOBF= 000400  D.GE   = 000000  FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131  C.NUM  = 000020  D.GT   = 000000  FA.DLK= 001000
A.NUM  = 000006          CHR.BS= 000010  C.OP   = 000004  D.IBM  = 000002  FA.ENB= 100000
A.PTR  = 000000          CHR.CC= 000003  C.OPN  = 010000  D.IND  = 000002  FA.EXC= 002000
A.TMP  = 000010          CHR.CR= 000015  C.RDLN= 177776  D.INNR= 000013  FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032  C.REWT= 002000  D.INTR= 000006  FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046  C.RND  = 000010  D.LE   = 000000  FA.NSR= 000200
BLZBUB  005116R     025  CHR.LF= 000012  C.SBIT= 177756  D.LOG  = 000001  FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043  C.SPFX= 000100  D.LT   = 000000  FA.RD  = 000001
CCNT    005072R     025  CHR.SI= 000017  C.SPGL= 000200  D.MAX  = 000000  FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057  C.SPLX= 000001  D.MEM  = 000002  FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016  C.SPRS= 000002  D.MIN  = 000000  FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011  C.SRC  = 177774  D.MUL  = 000000  FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106  C.SRCP= 000024  D.NE   = 000000  FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141  C.TT   = 000004  D.NN   = 000000  FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145  C.WIDC= 177770  D.NR   = 000000  FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163  C.WIDR= 177772  D.OR   = 000000  FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164  DEBUG  = 000000  D.PLUS= 000000  FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165  DEFERR= 104405  D.PSUB= 000006  FD.CR  = 000002
CHAN08  001766R     025  CHR..Z= 000172  DEVIL   005114R     025 D.PWR  = 000001  FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060  DEVNTF= 104470  D.QQAD= 000000  FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061  DIGIT   000136RG    020 D.QUAD= 000000  FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001  DIGITS  000006R     025 D.RES  = 000000  FD.INS= 000010
CHAN12  002676R     025  CMDBUF  003250R     025 DIMAX  = 000020  D.RHO  = 000000  FD.ISP= 002000
CHAN13  000022R     023  CMDMAX= 000010  DIMBLK  003112R     025 D.ROT  = 000003  FD.MNT= 100000
CHAN14  000204R     023  CMDNBR  000000RG    020 DIMPTR  003114R     025 D.SCON= 000006  FD.OSP= 004000
CHNOPN= 104471           CNTSIZ  005130R     025 DIMTYP  003116R     025 D.SDIF= 000006  FD.PLC= 000004
CHRAND= 000051           COMBIN  005076R     025 DIRFUL= 104466  D.SUB  = 000000  FD.PRN= 000004
CHRBEL= 000007           COMCNT  005126R     025 DOMERR= 104417  D.SUBS= 000006  FD.PSE= 010000
CHRBKR= 000133           COMERR= 104425  DPTERR= 104416  D.TAKE= 000002  FD.RAH= 000001
CHRCOL= 000200           COMPTR  005122R     025 DYDCNT  003224R     C25 D.TRAN= 000004  FD.RAN= 000002
CHRCOX= 000114           COMRTN  003156R     025 DYDTAB= ****** GX  D.UNON= 000006  FD.REC= 000001
CHRCMA= 000054           COMSKP  005124R     025 D.AND  = 000000  D.$MAX= 000000  FD.RWM= 000001
CHRCOL= 000076           CORPTR  005120R     025 D.BXCT= 000007  D.$MIN= 000000  FD.SDI= 000020
CHRDEL= 000107           CS.ASS= 000200  D.BXIO= 000007  D.$MUL= 000000  FD.SQD= 000040
CHRDIV= 000053           CS.BOO= 000010  D.BXNL= 000007  D.$PLU= 000000  FD.TTY= 000004
CHRDLR= 000176           CS.FLO= 000020  D.BXPP= 000007  D.$RES= 000000  FD.WBH= 000002
CHRDLT= 000110           CS.IDN= 000040  D.BXPW= 000007  D.$SUB= 000000  FF.CHR= 000005
CHRDOT= 000056           CS.ONE= 000100  D.BXRL= 000007  D.1COM= 000001  FF.NV  = 000003
CHREP  = 000105          CTEPS   000152R     025 D.CIRC= 000001  D.1EXP= 000002  FF.POE= 000002
CHRERR= 104431           CTMAX   000212R     025 D.COMB= 000001  D.1ROT= 000003  FF.RWD= 000001
CHRFIL= 000177           CTRMAX  000172R     025 D.COMP= 000001  EDICHR  005063R     025 FF.RWF= 000006
```

CMDNBR MACRO M1113 08-FEB-80 11:12 PAGE 47-2
SYMBOL TABLE

```
FF.SPC= 000004          F.FFBY= 000014      HASHTB  005140R 025 MAXCHN= 000014          M.RAVL= 000000
FHEADR  005012R 025     F.FLAG= 000002      HSHMSK= 177601      MONBLK  003120R 025 M.RCIP= 000000
FHRET   005040R 025     F.FLB = 000100      HSHSIZ= 000200      MONCVT  003132R 025 M.RDCT= 000011
FILNTF= 104467          F.FNAM= 000110      ICHAN   005050R 025 MONLEN  003130R 025 M.REVS= 000003
FIS    = 000000         F.FNB = 000102      ILFNAM= 104464      MONPTR  003122R 025 M.ROLL= 000000
FISCON= 046000          F.FNDF= 000010      ILLCHN= 104540      MONSFT  003126R 025 M.SATA= 000001
FIXPTR  003074R 025     F.FTYP= 000116      ILLSW = 104565      MONTAB= ****** GX       M.SCAN= 000011
FLAGS   004764R 025     F.FUN = 000010      INDERR= 104410      MONTYP  003124R 025 M.SHAP= 000000
FLCBUF  003250R 025     F.FVER= 000120      INNTMP  003250R 025 MONVAL  003134R 025 M.SIGN= 000000
FLINE   005014R 025     F.HEAD= 000006      INNTM2  005016R 025 M.ABS = 000000      M.SUBR= 000001
FLOEXP  003270R 025     F.HIBK= 000004      INPTR   000064R 025 M.BOL = 000200      M.SYM = 000203
FLONEG  003272R 025     F.LBL = 000020      INSNAM= 104462      M.BXAV= 000007      M.SYN = 000201
FLSWTB  000222R 025     F.LINE= 000012      IOERR = 104477      M.BXCR= 000007      M.TRAN= 000012
FLTLEN= 000010          F.LOC = 000004      I.CNT = 177776      M.BXCT= 000007      M.$ABS= 000000
FNINSI  005044R 025     F.LSTC= 000004      I.DIM = 177772      M.BXEX= 000007      M.$CEI= 000000
FORIGN  000032R 025     F.LUN = 000042      I.LEN = 000012      M.BXFX= 000007      M.$FLR= 000000
FO.APD= 000106          F.MBCT= 000054      I.OERR= 000062      M.BXIO= 000007      M.$IDN= 000000
FO.MFY= 000002          F.MBC1= 000055      I.OLD = 177770      M.BXLC= 000003      M.$NEG= 000000
FO.RD = 000001          F.MBFG= 000056      I.PTR = 177774      M.BXNC= 000007      M.$SIG= 000000
FO.UPD= 000006          F.MON = 000002      I.STR = 177766      M.BXNL= 000007      M.1RDT= 000011
FO.WRT= 000016          F.NIL = 000001      KILL   = ****** GX   M.BXPP= 000007      M.1REV= 000003
FPMODE  000042R 025     F.NRBD= 000024      LABERR= 104406      M.BXPW= 000007      M.1SCN= 000011
FPP    = 000000         F.NREC= 000030      LABLPT  005042R 025 M.BXRL= 000007      NAMBUF  003240R 025
FPPLEN= 000004          F.OPN = 000010      LARGBL  003162R 025 M.BXTS= 000007      NB.DEV= 000200
FREBLK  007400R 025     F.OVBS= 000030      LARGCV  003176R 025 M.BXWA= 000007      NB.DIR= 000100
FUZZ    000142R 025     F.PREV= 000000      LARGFL  003160R 025 M.CEI = 177777      NB.NAM= 000004
FUZZY   000114RG 020    F.PRNT= 100000      LARGLE  003174R 025 M.CHR = 000202      NB.SD1= 000400
FXPTR1  003076R 025     F.PROT= 010000      LARGLN  003166R 025 M.CMT = 000206      NB.SD2= 001000
F$EVAL= 000100          F.RACC= 000016      LARGPT  003164R 025 M.DOUT= 000006      NB.SNM= 000040
F$EVI = 000200          F.RATT= 000001      LARGSF  003172R 025 M.DQAD= 000006      NB.STP= 000020
F$EXEC= 000400          F.RCNM= 000034      LARGTY  003170R 025 M.ENCD= 000004      NB.SVR= 000010
F$FUNC= 001000          F.RCTL= 000017      LARGVA  003200R 025 M.EXE = 000000      NB.TYP= 000002
F.ACTL= 000076          F.READ= 004000      LBINC   005100R 025 M.EXP = 000001      NB.VER= 000001
F.ALOC= 000040          F.REDF= 000040      LBUFF   003316R 025 M.FACT= 000001      ND.DEV= 000200
F.BBFS= 000062          F.RES = 000040      LBUFLN= 000621      M.FIL = 000177      ND.EXT= 000002
F.BDB = 000070          F.RET = 177772      LCOUNT  005106R 025 M.FLR = 177777      ND.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002      LENERR= 104412      M.FMT = 000004      ND.PPN= 000100
F.BITS= 177774          F.RTYP= 000000      LENTAB  000054R 025 M.GDUP= 000003      ND.$EX= 000000
F.BKDN= 000026          F.R2  = 000014      LEXPTR  003100R 025 M.GDWN= 000003      ND.$ES= 000020
F.BKDS= 000020          F.R3  = 000016      LFTOP   005066R 025 M.GETI= 000006      ND.$NM= 000000
F.BKEF= 000050          F.R4  = 000020      LINC    005102R 025 M.IBM = 000003      ND.$NS= 000040
F.BKP1= 000051          F.SEQN= 000100      LINE    005016R 025 M.IDN = 000000      ND.$PN= 001400
F.BKST= 000024          F.SPDV= 000072      LININC  005026R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKVB= 000064          F.SPUN= 000074      LISIZE= 000020      M.IOTA= 000000      NEWLIN= ****** GX
F.BR  = 000020          F.STBK= 000036      LITOP   005072R 025 M.LIT = 000205      NGET   = ****** GX
F.CHR = 000075          F.STK = 000022      LNLNG = 104541      M.LN  = 000001      NINPT1  003302R 025
F.CNTG= 000034          F.STOP= 000004      LOCLCT  005036R 025 M.MMD = 000010      NINPT2  003300R 025
F.CODE= 000010          F.STP = 000001      LOWCOR  000002R 025 M.NEG = 000000      NINSTK  003304R 025
F.DFNB= 000046          F.SUSP= 000002      LSTERP  004766R 025 M.NOT = 000000      NINTMP  003310R 025
F.DSPT= 000044          F.TRC = 000002      LWIDTH  000070R 025 M.NUB = 000006      NINTYP  003306R 025
F.DVNM= 000134          F.UNIT= 000136      L.BITS= 000003      M.NUM = 000204      NOTAVL= 104472
F.EFBK= 000010          F.URBD= 000020      L.CNT = 000002      M.OUTR= 000012      NOTOPN= 104473
F.EFN = 000050          F.VAL = 177776      L.PRNT= 000001      M.PITM= 000000      NPRT1   004762R 025
F.END = 002000          F.VBN = 000064      L.PTR = 000000      M.PRDS= 000006      NULL    007364R 025
F.EOBB= 000032          F.VBSZ= 000060      L.PTR1= 000004      M.QQAD= 000000      NUMCHN= 000020
F.ERR = 000052          GNOME   003102R 025 L.SUBS= 000002      M.QUAD= 000000      NUMDFL  003230R 025
F.FACC= 000043          HASHC   005136R 025 MACHIN= 000106      M.RAND= 000012      NUMEFL  003233R 025
```

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 47-3
SYMBOL TABLE

NUMESN  003236R     025 ORIGIN  000030R     025 R.BITS= 000003          SYSTAT  000000R     024 T.LIST= 000400
NUMFLG  003230R     025 ORIGNN  000122RG    020 R.FIX = 000001          SY.BAD= 000001          T.LIT = 000020
NUMLEN  000062R     025 OUTPTR  000066R     025 R.LCT = 000002          SY.BRK= 100000          T.MON = 000010
NUMNEG  003232R     025 OVROPR= 000000          R.LN  = 000002          SY.BS = 000010          T.NIL = 000004
NUMQFL  003234R     025 PATCH   007344R     025 R.LNUM= 000010          SY.CLR= 000040          T.NUM = 002000
NUMRIG  003231R     025 PRECSN  000052R     025 R.NEXT= 000006          SY.ERP= 000020          T.OPFN= 000001
NUMWID  003226R     025 PRINTM= ****** GX       R.OFF = 000000          SY.NOU= 000100          T.OTHR= 040000
NUMZFL  003235R     025 RANDAC  003064R     025 R.PREV= 000004          SY.OPN= 000002          T.SOP = 000002
NYIERR= 104432          RANDY   003062R     025 R.PTR = 000000          SY.PMT= 001000          T.SPOP= 000040
N.BLSZ= 000036          RANERR= 104411          R.SEQ = 000003          SY.SRT= 000400          T.TMP = 000200
N.DESC= 000020          RARGBL  003204R     025 R.VAR = 000002          SY.SVC= 000004          T.VAL = 100000
N.DEV = 000032          RARGCV  003220R     025 R6    =%000006          SY.SVL= 000200          T.VAR = 001000
N.DEVN= 000034          RARGFL  003202R     025 R7    =%000007          S.AB  = 177776          UPPER = 177400
N.DID = 000024          RARGLE  003216R     025 SI    =%000005          S.ASB = 100000          UTNAME  000344R     023
N.DVNM= 000032          RARGLN  003210R     025 SIDAMN= 104433          S.BFHD= 000020          VALERR= 104413
N.EXT = 000014          RARGPT  003206R     025 SIFRAM  003314R   025 S.BITS= 000002          VARBLE  007400R     025
N.FID = 000000          RARGSF  003214R     025 SISTAK  006350R   025 S.CHR1= 000003          VARBLS  000000R     025
N.FNAM= 000006          RARGTY  003212R     025 SITOP   006346R   025 S.FATT= 000016          VERNUM  000000R     025
N.FTYP= 000014          RARGVA  003222R     025 SITP    007144R   025 S.FDB = 000140          VRSION= 000002
N.FVER= 000016          RBASE   005074R     025 SIWARN= 000200          S.FLB = 040000          WASL    000000RG    002
N.NAME= 000006          RCOUNR  005112R     025 SIZTAB  000044R   025 S.FNAM= 000006          WIDTH   000072R     025
N.NEXT= 000022          RCOUNT  005110R     025 SI.BIT= 177774          S.FNB = 000036          WIDTTH  000130RG    020
N.PPN = 000024          REDCNT  005070R     025 SI.LEN= 177772          S.FNBW= 000017          WROUND  000020R     025
N.STAT= 000020          REDIGN  005066R     025 SI.RET= 177772          S.FNTY= 000004          WSDATE  000136R     025
N.UNIT= 000034          REDOFF  005072R     025 SI.VAL= 177776          S.FTYP= 000002          WSFULL= 104403
OBUFLN= 001000          REENTR  003060R     025 SKPSIZ  005134R   025 S.LEN = 000002          WSNAME  000074R     025
OCHAN   005052R     025 RESBIT  003142R     025 SPHOLD  000004R   024 S.LINK= 177774          WSTIME  000132R     025
OMODE   000402R     023 RESCVT  003152R     025 SPINIT  000002R   024 S.NFEN= 000020          $FIS  = ****** G
OPTMP0  005066R     025 RESERR= 104440          STKINC= 000774          S.TYPE= 140000          $FPP  = ****** G
OPTMP1  005070R     025 RESLEN  003150R     025 STOP  = 104420          S.VAL = 000000          $FPPLN= ****** G
OPTMP2  005072R     025 RESPTR  003140R     025 SUBCNT  003104R   025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP3  005074R     025 RESSFT  003146R     025 SUBDCT  003106R   025 TKFRAM  005066R     025 $RSTS = ****** G
OPTMP4  005076R     025 RESTYP  003144R     025 SUBPTR  003110R   025 TKLEFT  005070R     025 $RSX  = ****** G
OPTMP5  005100R     025 RESULT  003136R     025 SYMB1   005344R   025 TMODE   000403R     023 $RT11 = ****** G
OPTMP6  005102R     025 RESVAL  003154R     025 SYMF    005340R   025 TMPTAG= 000166          .BOOLE= 000002
OPTMP7  005104R     025 RINC    005104R     025 SYMPT1  005346R   025 TTYBUF  004140R     025 .CHARA= 000000
OPTMP8  005106R     025 RLTBAS  005066R     025 SYMTAB  005352R   025 TTYLEN= 000622          .DEF. = 000002
OPTMP9  005110R     025 ROTINC  005070R     025 SYMTBF= 104404          TURKEY= 104436          .FLOAT= 000006
OPTM10  005112R     025 ROUND   000010R     025 SYMTOP  005350R   025 TYPE  = 000166          .FSRCB= ****** G
OPTM14  005026R     025 ROVER   003312R     025 SYMTP   006272R   025 T.BOL = 020000          .INTEG= 000004
OP..CT= ****** GX       RSTS  = 000000          SYMWRN= 000054          T.CON = 004000          .MOLUN  000404R     023
OP..IO= ****** GX       RSX   = 000115          SYNERR= 104407          T.DIM = 000100          ...GBL= 000000
OP..PP= ****** GX       RTOP    005070R     025 SYSERR= 104402          T.FUN = 010000          ...TPC= 000020
OP..PW= ****** GX       RT11  = 000000

. ABS.  000000      000
        000000      001
TABLES  000006      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
```

```
CMDNBR  MACRO M1113  08-FEB-80 11:12  PAGE 47-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000144      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12973 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:57
CMDNBR,DM1:[305,301]CMDNBR/-SP=APLRX1,APLDEF,CMDNBR,END
```

FILEID**CMPRSS

```
  CCCCCCCC    MM       MM    PPPPPPPP      RRRRRRRR        SSSSSSSS      SSSSSSSS
  CCCCCCCC    MM       MM    PPPPPPPP      RRRRRRRR        SSSSSSSS      SSSSSSSS
CC            MMMM   MMMM    PP      PP    RR      RR    SS            SS
CC            MMMM   MMMM    PP      PP    RR      RR    SS            SS
CC            MM  MM  MM    PP      PP    RR      RR    SS            SS
CC            MM  MM  MM    PP      PP    RR      RR    SS            SS
CC            MM      MM    PPPPPPPP      RRRRRRRR        SSSSSS        SSSSSS
CC            MM      MM    PPPPPPPP      RRRRRRRR        SSSSSS        SSSSSS
CC            MM      MM    PP            RR  RR                  SS            SS
CC            MM      MM    PP            RR  RR                  SS            SS
CC            MM      MM    PP            RR    RR                SS            SS        ....
CC            MM      MM    PP            RR    RR                SS            SS        ....
  CCCCCCCC    MM      MM    PP            RR      RR    SSSSSSSS    SSSSSSSS        ....
  CCCCCCCC    MM      MM    PP            RR      RR    SSSSSSSS    SSSSSSSS        ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:14
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 78, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    CMPRSS           ROUTINE TO FORM 2-D STRUCTURE FROM VECTOR
```

```
CMPRSS  MACRO M1113  08-FEB-80 11:14  PAGE 1


     1          000000                  RT11    =  0            ;NOT RT-11
     2          000000                  RSTS    =  0            ;NOT RSTS/E
     3          000115                  RSX     = 'M            ;RSX-11/M
     4
     5          000000                  EIS     =  0            ;NO EIS
     6          000000                  FIS     =  0            ;NO FIS
     7          000000                  FPP     =  0            ;NO FPP
     8
     9          000004                  FPPLEN  =  4.           ;SINGLE PRECISION
    10
    11          000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CMPRSS  MACRO M1113  08-FEB-80 11:14  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CMPRSS
                                .IDENT  \V02.03\

                                .SBTTL  CMPRSS          ROUTINE TO FORM 2-D STRUCTURE FROM VECTOR

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This Routine is Used by .BXCR, Monadic Thorn, etc...
                                ;       It takes a Vector and Converts it into a 2-Dimensional
                                ;       Structure, removing <CR><LF> pairs, etc.
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    CMPRSS
                                ;
                                ; IMPLICIT INPUTS:
                                ;       On the Top of the SI    - A Character Vector to be re-formatted
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       On the Top of the SI    - A new Character Array/Vector
                                ;
                                ; SIDE EFFECTS:
                                ;       The old character vector may be deleted and replaced
                                ;--
000000                                  .PSECT  CODE

000000                          CMPRSS::
000000                                  CALL    CARVE           ;Carve Up Argument
000004  016501  177776                  MOV     SI.VAL(SI),R1   ;Point to the Argument
000010  011104                          MOV     (R1),R4         ;Point Specifically to the String
000012  016103  000004                  MOV     A.ALEN(R1),R3   ;Get the Total Length (Bytes)
000016  005000                          CLR     R0              ;The Current Line Width
000020  005046                          CLR     -(SP)           ;Count of # of Lines
000022  005046                          CLR     -(SP)           ;The Maximum Line Width
                                ;
                                ; Loop for Each Line - find the Specs (# of Lines, Max Width)
                                ;
000024  005001                  1$:     CLR     R1              ;Clear <CR><LF> Flag
000026                          2$:     GETC    4$              ;Get a Character for Processing
000026  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000030  002416                          BLT 4$                  ;NO, BRANCH TO 4$
000032  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000034  005200                  3$:     INC     R0              ;Count the Character
000036  120227  000015                  CMPB    R2,#CHR.CR      ;Check for <CR>
000042  001370                          BNE     1$              ;Nope, try again
000044                                  GETC    4$              ;Get another Character
000044  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000046  002407                          BLT 4$                  ;NO, BRANCH TO 4$
000050  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000052  120227  000012                  CMPB    R2,#CHR.LF      ;Check for <LF>
000056  001366                          BNE     3$              ;No, check for <CR>
000060  005300                          DEC     R0              ;Forget the <CR> from the Count
000062  005701                          TST     R1              ;<CR><LF> Flag set?
000064  001360                          BNE     2$              ;Yes, so look for some more EOL pairs
000066  020016                  4$:     CMP     R0,(SP)         ;Which is Bigger
000070  002401                          BLT     5$              ;The Old Value is...
000072  010016                          MOV     R0,(SP)         ;No, use New Value
000074  005000                  5$:     CLR     R0              ;Starting New Line
```

```
CMPRSS  MACRO M1113  08-FEB-80 11:14  PAGE 44-1
CMPRSS          ROUTINE TO FORM 2-D STRUCTURE FROM VECTOR

     52 000076  005701                          TST     R1              ;Have we found Previous <CR><LF> Pairs?
     53 000100  001003                          BNE     6$              ;Yes, so Skip this One
     54 000102  005266  000002                  INC     2(SP)           ;Add Another Line
     55 000106  005201                          INC     R1              ;Set <CR><LF> Flag
     56 000110  005703                  6$:     TST     R3              ;Anything Left?
     57 000112  003345                          BGT     2$              ;Loop For All
     58                                         ;
     59                                         ; Well, We've Finished Accumulating Info. If Vector, use
     60                                         ; old value (Forget end <CR><LF>, if any)
     61                                         ;
     62 000114  016601  000002                  MOV     2(SP),R1        ;Get the number of lines
     63 000120  022701  000001                  CMP     #1.,R1          ;Is there only one Line?
     64 000124  001011                          BNE     7$              ;Is a Vector, Use old Value
     65 000126                                  POP     R0              ;Get the Max Line Width
     66 000130  016501  177776                  MOV     SI.VAL(SI),R1   ;Pointer to Data
     67 000134  010061  000004                  MOV     R0,A.ALEN(R1)   ;Total Length
     68 000140  010061  000006                  MOV     R0,A.DIM1(R1)   ;   and Length of Vector
     69 000144  005726                          TST     (SP)+           ;Clean up garbage on stack
     70 000146                                  RETURN                  ;Finished !
     71                                         ;
     72                                         ; Well, I guess we really have to do the allocate and copy.
     73                                         ; DAMN!
     74                                         ;
     75 000150  011600                  7$:     MOV     (SP),R0         ;Get the maximum line width
     76 000152                                  MUL     R0,R1           ;Compute Real Size (Characters)
     77 000164  103524                          BCS     16$             ;Too Large, Error
     78 000166  012700  000006                  MOV     #A.DIM1,R0      ;Get overhead for Header
     79 000172  016502  177770                  MOV     SI.LEN+SI.VAL(SI),R2  ;Pointer to original source argument
     80 000176  116203  000002                  MOVB    A.DIMS(R2),R3   ;Get the # of dimensions
     81 000202                                  CONINC  R3              ;Cheat for scalars
     82 000206  010304                          MOV     R3,R4           ;Copy the # of dimensions
     83 000210  006303                          ASL     R3              ;Make it words
     84 000212  060300                          ADD     R3,R0           ;Combine with other overhead
     85 000214  010003                          MOV     R0,R3           ;Save data offset
     86 000216  060100                          ADD     R1,R0           ;Combine with data space required
     87 000220                                  CALL    GETCOR          ;Get the Space
     88 000224                                  SIFRAM  ,#T.VAL         ;Build SI Parse Value Frame
     89 000236  010010                          MOV     R0,(R0)         ;Fill in array block header
     90 000240  060320                          ADD     R3,(R0)+        ;  Add in data offset
     91 000242  110420                          MOVB    R4,(R0)+        ;  DIMS
     92 000244  112720  000200                  MOVB    #<A.ASB+.CHARACTER>,(R0)+  ;Character Array
     93 000250  010120                          MOV     R1,(R0)+        ;  ALEN
     94 000252  022704  000001                  CMP     #1,R4           ;  Result a vector?
     95 000256  001406                          BEQ     9$              ;  Yes, Skip copying the dimensions
     96 000260  005304                          DEC     R4              ;  Only copy the first n-1 dimensions
     97 000262  062702  000006                  ADD     #A.DIM1,R2      ;  Point to the source dimension list
     98 000266  012220                  8$:     MOV     (R2)+,(R0)+     ;  Copy the dimensions
     99 000270                                  SOB     R4,8$           ;  Loop through the list
    100 000274  011620                  9$:     MOV     (SP),(R0)+      ;  Fill in last dim (Length of Each Line)
    101 000276  016502  177770                  MOV     SI.LEN+SI.VAL(SI),R2  ;Point to Old Data
    102 000302  016203  000004                  MOV     A.ALEN(R2),R3   ;Get the Length
    103 000306  011204                          MOV     (R2),R4         ;   and the Pointer
    104 000310  005001                          CLR     R1              ;Clear the <CR><LF> Flag
    105 000312                                  PUSH    (SP)            ;Copy the line width
    106 000314  000401                          BR      11$             ;Start the presses
    107                                         ;
    108                                         ; OK, Now let's do the Transfer Loop
```

CMPRSS ROUTINE TO FORM 2-D STRUCTURE FROM VECTOR

```
                                                ;
000316  005316                          10$:    DEC     (SP)            ;Decrease count of what's left on the line
000320                                  11$:    GETC    13$             ;Get a Character
000320  005303                                  DEC R3                          ;ANY CHARACTERS LEFT?
000322  002417                                  BLT 13$                 ;NO, BRANCH TO 13$
000324  112402                                  MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000326  120227  000015                          CMPB    R2,#CHR.CR      ;is it a <CR>?
000332  001010                                  BNE     12$             ;Nope
000334  121427  000012                          CMPB    (R4),#CHR.LF    ;Followed By a <LF>?
000340  001005                                  BNE     12$             ;Nope
000342                                          GETC    13$             ;Adjust for <LF>
000342  005303                                  DEC R3                          ;ANY CHARACTERS LEFT?
000344  002406                                  BLT 13$                 ;NO, BRANCH TO 13$
000346  112402                                  MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000350  005201                                  INC     R1              ;Set the <CR><LF> Flag
000352  000762                                  BR      11$             ;Go look for some <CR><LF> pairs
000354  110220                          12$:    MOVB    R2,(R0)+        ;Transfer the character
000356  005701                                  TST     R1              ;Is the <CR><LF> Flag set?
000360  001756                                  BEQ     10$             ;No, Not at the end of a line
000362                                  13$:    POP     R2              ;Any Space Left on the Line?
000364  003404                                  BLE     15$             ;Nope, skip Fill
000366  112720  000040                  14$:    MOVB    #CHRSPA,(R0)+   ;Move in a Space
000372                                          SOB     R2,14$          ;And Loop to Finish
000376                                  15$:    PUSH    (SP)            ;Restore Count of Remaining
000400  005001                                  CLR     R1              ;Clear the <CR><LF> Flag
000402  005703                                  TST     R3              ;Any more characters left?
000404  003344                                  BGT     10$             ;Loop for New Lines
000406  016516  177776                          MOV     SI.VAL(SI),(SP) ;Save New Data Pointer
000412  016565  177770  177776                  MOV     SI.VAL+SI.LEN(SI),SI.VAL(SI) ;Copy Old Data Pointer
000420                                          POP     SI.VAL+SI.LEN(SI) ;Restore Data Pointer
000424                                          CALL    KILL            ;Die, Die, Die
000430  005726                                  TST     (SP)+           ;Skip Garbage
000432  005726                                  TST     (SP)+           ;All of it
000434                                          RETURN                  ;Finished!
                                                ;
000436  104403                          16$:    WSFULL                  ;Workspace cannot hold this
```

```
CMPRSS  MACRO M1113  08-FEB-80 11:14  PAGE 46
CMPRSS          ROUTINE TO FORM 2-D STRUCTURE FROM VECTOR

      1         000001                          .END
```

CMPRSS MACRO M1113 08-FEB-80 11:14 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    C.ALOC= 000162    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLBR= 000073    C.BLOK= 000064    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRLPR= 000072    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMNS= 000137    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRMUL= 000075    C.BUFL= 177762    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRNEG= 000100    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPDL= 000271    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPLU= 000055    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRPWR= 000120    C.DIG = 000040    D.EXPD= 000002    ERRO    004774R     025
A.ALEN= 000004           CHRQST= 000121    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRQUO= 000113    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R     025
A.ASB = 000200           CHRRBR= 000047    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.BITS= 000003           CHRRPR= 000042    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.CON = 000020           CHRSEM= 000074    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSMO= 000112    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIM1= 000006           CHRSPA= 000040    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.NUM = 000006           CHRUPA= 000131    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.PTR = 000000           CHR.BS= 000010    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.TMP = 000010           CHR.CC= 000003    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
BADDIR= 104463           CHR.CR= 000015    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADERR= 104437           CHR.CZ= 000032    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BLZBUB  005116R     025  CHR.GT= 000046    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LF= 000012    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
CARVE = ****** GX        CHR.LT= 000043    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVIL   005114R     025 D.QUAD= 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R     025 D.RHO = 000000    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMBLK  003112R     025 D.SCON= 000006    FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010    DIMPTR  003114R     025 D.SDIF= 000006    FD.PLC= 000004
CHAN14  000204R     023  CMPRSS  000000RG     020 DIMTYP  003116R     025 D.SUB = 000000    FD.PRN= 000004
CHNOPN= 104471           CNTSIZ  005130R     025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051           COMBIN  005076R     025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007           COMCNT  005126R     025 DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133           COMERR= 104425    DYDCNT  003224R     025 D.UNON= 000006    FD.REC= 000001
CHRBOL= 000200           COMPTR  005122R     025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114           COMRTN  003156R     025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054           COMSKP  005124R     025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076           CORPTR  005120R     025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.ASS= 000200        D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.BOO= 000010        D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176           CS.FLO= 000020        D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110           CS.IDN= 000040        D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056           CS.ONE= 000100        D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105           CTEPS   000152R     025 D.COMP= 000001    EDICHR  005063R     025 FF.RWF= 000006
CHRERR= 104431           CTMAX   000212R     025 D.CONT= 000006    EDICNT  005064R     025 FF.SPC= 000004
CHRFIL= 000177           CTRMAX  000172R     025 D.CTLM= 000000    EDIFLG  005062R     025 FHEADR  005012R     025
```

CMPRSS MACRO M1113 08-FEB-80 11:14 PAGE 46-2
SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110        HSHSIZ= 000200          MONBLK  003120R     025 M.RDCT= 000011
FILNTF= 104467          F.FNB = 000102        ICHAN   005050R     025 MONCVT  003132R     025 M.REVS= 000003
FIS   = 000000          F.FNDF= 000010        ILFNAM= 104464          MONLEN  003130R     025 M.ROLL= 000000
FISCON= 046000          F.FTYP= 000116        ILLCHN= 104540          MONPTR  003122R     025 M.SATA= 000001
FIXPTR  003074R     025 F.FUN = 000010        ILLSW = 104565          MONSFT  003126R     025 M.SCAN= 000011
FLAGS   004764R     025 F.FVER= 000120        IMUL  = ****** GX       MONTYP  003124R     025 M.SHAP= 000000
FLCBUF  003250R     025 F.HEAD= 000006        INDERR= 104410          MONVAL  003134R     025 M.SIGN= 000000
FLINE   005014R     025 F.HIBK= 000004        INNTMP  003250R     025 M.ABS = 000000          M.SUBR= 000001
FLOEXP  003270R     025 F.LBL = 000020        INNTM2  005016R     025 M.BOL = 000200          M.SYM = 000203
FLONEG  003272R     025 F.LINE= 000012        INPTR   000064R     025 M.BXAV= 000007          M.SYN = 000201
FLSWTB  000222R     025 F.LOC = 000004        INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FLTLEN= 000010          F.LSTC= 000004        IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FNINSI  005044R     025 F.LUN = 000042        I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FORIGN  000032R     025 F.MBCT= 000054        I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055        I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056        I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.RD = 000001          F.MON = 000002        I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001        I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024        I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FPMODE  000042R     025 F.NREC= 000030        KILL  = ****** GX       M.BXPW= 000007          M.1SCN= 000011
FPP   = 000000          F.OPN = 000010        LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FPPLEN= 000004          F.OVBS= 000030        LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FREBLK  007400R     025 F.PREV= 000000        LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FUZZ    000142R     025 F.PRNT= 100000        LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FXPTR1  003076R     025 F.PPOT= 010000        LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016        LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001        LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034        LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017        LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000        LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040        LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F.BBFS= 000062          F.RES = 000040        LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.BDB = 000070          F.RET = 177772        LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002        LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000        LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BKDN= 000026          F.R2  = 000014        LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BKDS= 000020          F.R3  = 000016        LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
F.BKEF= 000050          F.R4  = 000020        LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKP1= 000051          F.SEQN= 000100        LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
F.BKST= 000024          F.SPDV= 000072        LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKVB= 000064          F.SPUN= 000074        LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BR  = 000020          F.STBK= 000036        LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.CHR = 000075          F.STK = 000022        LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.CNTG= 000034          F.STOP= 000004        LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.CODE= 000010          F.STP = 000001        LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.DFNB= 000046          F.SUSP= 000002        LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.DSPT= 000044          F.TRC = 000002        LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
F.DVNM= 000134          F.UNIT= 000136        LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.EFBK= 000010          F.URBD= 000020        LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.EFN = 000050          F.VAL = 177776        L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.END = 002000          F.VBN = 000064        L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EOBB= 000032          F.VBSZ= 000060        L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.ERR = 000052          GETCOR= ****** GX     L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.FACC= 000043          GNOME   003102R     025 L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.FFBY= 000014          HASHC   005136R     025 L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.FLAG= 000002          HASHTB  005140R     025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FLB = 000100          HSHMSK= 177601          MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
```

CMPRSS MACRO M1113 08-FEB-80 11:14 PAGE 46-3
SYMBOL TABLE

```
NUMQFL  003234R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BRK= 100000          T.LIST= 000400
NUMRIG  003231R     025 RANERR= 104411          R.OFF = 000000          SY.BS = 000010          T.LIT = 000020
NUMWID  003226R     025 RARGBL  003204R     025 R.PREV= 000004          SY.CLR= 000040          T.MON = 000010
NUMZFL  003235R     025 RARGCV  003220R     025 R.PTR = 000000          SY.ERP= 000020          T.NIL = 000004
NYIERR= 104432          RARGFL  003202R     025 R.SEQ = 000003          SY.NOU= 000100          T.NUM = 002000
N.BLSZ= 000036          RARGLE  003216R     025 R.VAR = 000002          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLN  003210R     025 R6    =%000006          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGPT  003206R     025 R7    =%000007          SY.SRT= 000400          T.SOP = 000002
N.DEVN= 000034          RARGSF  003214R     025 SI    =%000005          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGTY  003212R     025 SIDAMN= 104433          SY.SVL= 000200          T.TMP = 000200
N.DVNM= 000032          RARGVA  003222R     025 SIFRAM  003314R   025 S.AB  = 177776          T.VAL = 100000
N.EXT = 000014          RBASE   005074R     025 SISTAK  006350R   025 S.ASB = 100000          T.VAR = 001000
N.FID = 000000          RCOUNR  005112R     025 SITOP   006346R   025 S.BFHD= 000020          UPPER = 177400
N.FNAM= 000006          RCOUNT  005110R     025 SITP    007144R   025 S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          REDCNT  005070R     025 SIWARN= 000200          S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDIGN  005066R     025 SIZTAB  000044R   025 S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDOFF  005072R     025 SI.BIT= 177774          S.FDB = 000140          VARBLS  000000R     025
N.NEXT= 000022          REENTR  003060R     025 SI.LEN= 177772          S.FLB = 040000          VERNUM  000000R     025
N.PPN = 000024          REGTYP= 000001          SI.RET= 177772          S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESBIT  003142R     025 SI.VAL= 177776          S.FNB = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESCVT  003152R     025 SKPSIZ  005134R   025 S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESERR= 104440          SPHOLD  000004R   024 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESLEN  003150R     025 SPINIT  000002R   024 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESPTR  003140R     025 STKINC= 000774          S.LEN = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESSFT  003146R     025 STOP  = 104420          S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESTYP  003144R     025 SUBCNT  003104R   025 S.NFEN= 000020          $FIS  = ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBDCT  003106R   025 S.TYPE= 140000          $FPP  = ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBPTR  003110R   025 S.VAL = 000000          $FPPLN= ****** G
OPTMP4  005076R     025 RINC    005104R     025 SYMB1   005344R   025 TKDIMS  005072R   025 $MACHN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMF    005340R   025 TKFRAM  005066R   025 $RSTS = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMPT1  005346R   025 TKLEFT  005070R   025 $RSX  = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMTAB  005352R   025 TMODE   000403R   023 $RT11 = ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMTBF= 104404          TMPTAG= 000166          .BOOLE= 000002
OPTMP9  005110R     025 RSTS  = 000000          SYMTOP  005350R   025 TTYBUF  004140R   025 .CHARA= 000000
OPTM10  005112R     025 RSX   = 000115          SYMTP   006272R   025 TTYLEN= 000622          .DEF. = 000002
OPTM14  005026R     025 RTOP    005070R     025 SYMWRN= 000054          TURKEY= 104436          .FLOAT= 000006
ORIGIN  000030R     025 RT11  = 000000          SYNERR= 104407          TYPE  = 000166          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003          SYSER = ****** GX       T.BOL = 020000          .INTEG= 000004
OVROPR= 000000          R.FIX = 000001          SYSERR= 104402          T.CON = 004000          .MOLUN  000404R     023
PATCH   007344R     025 R.LCT = 000002          SYSTAT  000000R   024 T.DIM = 000100          ...GBL= 000000
PRECSN  000052R     025 R.LN  = 000002          SY.BAD= 000001          T.FUN = 010000          ...TPC= 000020
RANDAC  003064R     025 R.LNUM= 000010

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
```

```
CMPRSS  MACRO M1113  08-FEB-80 11:14  PAGE 46-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000440      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13004 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:57
CMPRSS,DM1:[305,301]CMPRSS/-SP=APLRX1,APLDEF,CMPRSS,END
```

FILEID**CMPTRT

```
  CCCCCCCC   MM      MM   PPPPPPPP     TTTTTTTTTT   RRRRRRRR     TTTTTTTTTT
  CCCCCCCC   MM      MM   PPPPPPPP     TTTTTTTTTT   RRRRRRRR     TTTTTTTTTT
CC           MMMM  MMMM   PP      PP       TT       RR      RR      TT
CC           MMMM  MMMM   PP      PP       TT       RR      RR      TT
CC           MM  MM  MM   PP      PP       TT       RR      RR      TT
CC           MM  MM  MM   PP      PP       TT       RR      RR      TT
CC           MM      MM   PPPPPPPP         TT       RRRRRRRR        TT
CC           MM      MM   PPPPPPPP         TT       RRRRRRRR        TT
CC           MM      MM   PP               TT       RR  RR          TT
CC           MM      MM   PP               TT       RR  RR          TT
CC           MM      MM   PP               TT       RR    RR        TT          ....
CC           MM      MM   PP               TT       RR    RR        TT          ....
  CCCCCCCC   MM      MM   PP               TT       RR      RR      TT          ....
  CCCCCCCC   MM      MM   PP               TT       RR      RR      TT          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:16
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS               RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CMPTRT           COMPUTE RESULT TYPE FOR OP
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 1


     1          000000                  RT11    =  0            ;NOT RT-11
     2          000000                  RSTS    =  0            ;NOT RSTS/E
     3          000115                  RSX     = 'M            ;RSX-11/M
     4
     5          000000                  EIS     =  0            ;NO EIS
     6          000000                  FIS     =  0            ;NO FIS
     7          000000                  FPP     =  0            ;NO FPP
     8
     9          000004                  FPPLEN  =  4.           ;SINGLE PRECISION
    10
    11          000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CMPTRT
                                .IDENT  \V02.03\

                                .SBTTL  CMPTRT          COMPUTE RESULT TYPE FOR OP

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;         This routine computes the intermediate source, intermediate
                                ;       result, and result data types for Monadic and Dyadic Scalar
                                ;       operators.
                                ;
                                ; INPUTS:
                                ;       R0      - Bits from 'DYDBTL' or 'MONBTL'
                                ;       R2      - (Largest) Source Type
                                ;       R4      - Operator Number
                                ;
                                ; IMPLICIT INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R0      - Result Intermediate Type  * 4
                                ;       R1      - Result Type
                                ;       R2      - Source Intermediate Type * 4
                                ;       R4      - Operator Number
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       COMRTN  - Set as pointer to Compare Routine
                                ;       RESTYP  - Set with the Result Type
                                ;       RESSFT  - Result shift count
                                ;       RESLEN  - Result element length
                                ;       RESCVT  - Set with pointer to Result store routine
                                ;
                                ; SIDE EFFECTS:
                                ;       NONE
                                ;--

000000                          .PSECT  CODE

000000                          CMPTRT::
000000  010001                          MOV     R0,R1           ;get bits
000002  042701  177771                  BIC     #^C<A.NUM>,R1   ;setup for Case Stmt
000006  060107                          ADD     R1,PC           ;Go to it

000010  000403                          BR      1$              ;comparison op
000012  000411                          BR      2$              ;Boolean Op
000014  000413                          BR      3$              ;Integer/Floating Op
000016  000431                          BR      6$              ;Floating Op

                                ;
                                ; Comparison Operation -                .LT .LE .EQ .GE .GT .NE
                                ;       Result, Result Intermediate = .BOOLEAN
                                ;       Source Intermediate = R2
                                ;
000020  012701  000002          1$:     MOV     #.BOOLEAN,R1    ;Result Type is Boolean
000024  016237  000000G 003156'         MOV     COMCT(R2),COMRTN;Pointer to Compare Routine
000032  000240                          NOP                     ;Just in case need MOV R2,R0
000034  000437                          BR      9$              ;Was 10$$
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 44-1
CMPTRT          COMPUTE RESULT TYPE FOR OP

                                        ;
                                        ; Boolean Operation -                   .AN .OR .NN .NR .NT
                                        ;       Result, Result Intermediate, Source Intermediate = .BOOLEAN
                                        ;
000036  005702                  2$:    TST     R2              ;Source Literal?
000040  001461                         BEQ     11$             ;Yes, DOMAIN ERROR
000042  000433                         BR      8$
                                        ;
                                        ; Integer/Floating Operation -          + - x .FL .CE .AB ?
                                        ;       This one is non-obvious to me, and I wrote it!
                                        ;
000044  005702                  3$:    TST     R2              ;Source Literal?
000046  001456                         BEQ     11$             ;Yes, DOMAIN ERROR
000050  020102                         CMP     R1,R2           ;Is source Integer or Boolean?
000052  002404                         BLT     4$              ;no, special for Float
000054  032700  000010                 BIT     #CS.BOOL,R0     ;How about the Result?
000060  001424                         BEQ     8$              ;Can't Be Boolean
000062  000414                         BR      7$              ;Yes, it can

000064  032700  000020          4$:    BIT     #CS.FLOAT,R0    ;Do we want to try an Integer result?
000070  001001                         BNE     5$              ;Yes, do so
000072  010201                         MOV     R2,R1           ;No, result is .FLOAT
000074  010200                  5$:    MOV     R2,R0           ;Source Intermediate is Float
000076  005404                         NEG     R4              ;Operator Number negated
000100  000416                         BR      10$
                                        ;
                                        ; Floating Operation -                  % * .LG .LO !
                                        ;       Result is .FLOAT (or maybe .BOOLEAN - special case)
                                        ;       Result Intermediate, Source Intermediate = .FLOAT
                                        ;
000102  005702                  6$:    TST     R2              ;Literal Source?
000104  001437                         BEQ     11$             ;Yes, DOMAIN ERROR
000106  032700  000010                 BIT     #CS.BOOL,R0     ;Weird Case of Boolean Sources
000112  001407                         BEQ     8$              ;Nope, no special bit
000114  020227  000002          7$:    CMP     R2,#.BOOLEAN    ;Boolean Sources?
000120  001004                         BNE     8$              ;Nope
000122  010100                         MOV     R1,R0           ;Result Intermediate
000124  010201                         MOV     R2,R1           ;Result is Boolean
000126  010002                         MOV     R0,R2           ;Source Intermediate is FLOAT
000130  000402                         BR      10$

000132  010102                  8$:    MOV     R1,R2           ;Source Intermediate

000134  010100                  9$:    MOV     R1,R0           ;Result Intermediate

000136  006302                  10$:   ASL     R2              ;Source Intermediate
000140  006302                         ASL     R2              ;  times 4
000142  006300                         ASL     R0              ;Result Intermediate
000144  006300                         ASL     R0              ;  times 4
000146  010137  003144'                MOV     R1,RESTYP       ;Save result type
000152  016137  000044' 003146'        MOV     SIZTAB(R1),RESSFT ;RESULT SHIFT CONSTANT
000160  016137  000054' 003150'        MOV     LENTAB(R1),RESLEN ;RESULT LENGTH CONSTANT
000166  060001                         ADD     R0,R1           ;Compute array pointer
000170  016137  000000G 003152'        MOV     RESCT(R1),RESCVT;Save pointer to Store Routine
000176  013701  003144'                MOV     RESTYP,R1       ;Restore Result Type
000202                                  RETURN                  ;Finished!
```

```
    115 000204  104417                  11$:    DOMERR                  ;Arrrrrgggggggg
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 46
CMPTRT           COMPUTE RESULT TYPE FOR OP

      1          000001                          .END
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.ALOC= 000162    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLPR= 000072    C.BLOK= 000064    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRMNS= 000137    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMUL= 000075    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRNEG= 000100    C.BUFL= 177762    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRPDL= 000271    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPLU= 000055    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPWR= 000120    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRQST= 000121    C.DIG = 000040    D.EXPD= 000002    ERR0    004774R     025
A.ALEN= 000004           CHRQUO= 000113    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRRBR= 000047    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R     025
A.ASB = 000200           CHRRPR= 000042    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.BITS= 000003           CHRSEM= 000074    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.CON = 000020           CHRSMO= 000112    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSPA= 000040    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIM1= 000006           CHRUPA= 000131    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.NUM = 000006           CHR.BS= 000010    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.PTR = 000000           CHR.CC= 000003    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.TMP = 000010           CHR.CR= 000015    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
BADDIR= 104463           CHR.CZ= 000032    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADERR= 104437           CHR.GT= 000046    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BLZBUB  005116R     025  CHR.LF= 000012    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LT= 000043    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
CCNT    005072R     025  CHR.SI= 000017    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN04  001056R     025  CHR..E= 000145    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN05  001240R     025  CHR..S= 000163    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN06  001422R     025  CHR..T= 000164    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN07  001604R     025  CHR..U= 000165    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN08  001766R     025  CHR..Z= 000172    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN09  002150R     025  CHR..0= 000060    DEVIL   005114R 025 D.QUAD= 000000    FD.F11= 040000
CHAN10  002332R     025  CHR..1= 000061    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN11  002514R     025  CH.AND= 000001    DIGITS  000006R 025 D.RHO = 000000    FD.ISP= 002000
CHAN12  002676R     025  CMDBUF  003250R 025 DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN13  000022R     023  CMDMAX= 000010    DIMBLK  003112R 025 D.SCON= 000006    FD.OSP= 004000
CHAN14  000204R     023  CMPTRT  000000RG 020 DIMPTR  003114R 025 D.SDIF= 000006    FD.PLC= 000004
CHNOPN= 104471           CNTSIZ  005130R 025 DIMTYP  003116R 025 D.SUB = 000000    FD.PRN= 000004
CHRAND= 000051           COMBIN  005076R 025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRBEL= 000007           COMCNT  005126R 025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBKR= 000133           COMCT = ****** GX   DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBOL= 000200           COMERR= 104425      DYDCNT  003224R 025 D.UNON= 000006    FD.REC= 000001
CHRBOX= 000114           COMPTR  005122R 025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRCMA= 000054           COMRTN  003156R 025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCOL= 000076           COMSKP  005124R 025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRDEL= 000107           CORPTR  005120R 025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDIV= 000053           CS.ASS= 000200      D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDLR= 000176           CS.BOO= 000010      D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLT= 000110           CS.FLO= 000020      D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDOT= 000056           CS.IDN= 000040      D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHREP = 000105           CS.ONE= 000100      D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRERR= 104431           CTEPS   000152R 025 D.COMP= 000001    EDICHR  005063R 025 FF.RWF= 000006
CHRFIL= 000177           CTMAX   000212R 025 D.CONT= 000006    EDICNT  005064R 025 FF.SPC= 000004
CHRFRO= 000135           CTRMAX  000172R 025 D.CTLM= 000000    EDIFLG  005062R 025 FHEADR  005012R     025
```

CMPTRT MACRO M1113 08-FEB-80 11:16 PAGE 46-2
SYMBOL TABLE

```
FHRET   005040R     025 F.FNAM= 000110      ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102      ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010      ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116      ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FIXPTR  003074R     025 F.FUN = 000010      INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R     025 F.FVER= 000120      INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R     025 F.HEAD= 000006      INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
FLINE   005014R     025 F.HIBK= 000004      INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R     025 F.LBL = 000020      INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R     025 F.LINE= 000012      IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R     025 F.LOC = 000004      I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004      I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R     025 F.LUN = 000042      I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055      I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056      I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002      I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FO.WRT= 000016          F.NRBD= 000024      LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R     025 F.NREC= 000030      LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FPP   = 000000          F.OPN = 000010      LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030      LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R     025 F.PREV= 000000      LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R     025 F.PRNT= 100000      LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R     025 F.PROT= 010000      LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016      LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001      LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034      LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017      LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000      LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040      LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040      LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772      LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002      LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000      LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014      LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016      LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020      LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BKP1= 000051          F.SEQN= 000100      LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.BKST= 000024          F.SPDV= 000072      LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.BKVB= 000064          F.SPUN= 000074      LITOP   005072R     025 M.LN  = 000001          NINSTK  003304R     025
F.BR  = 000020          F.STBK= 000036      LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CHR = 000075          F.STK = 000022      LOCLCT  005036R     025 M.NEG = 000000          NINTYP  003306R     025
F.CNTG= 000034          F.STOP= 000004      LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001      LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002      LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.ERR = 000052          GNOME   003102R     025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FACC= 000043          HASHC   005136R     025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
F.FFBY= 000014          HASHTB  005140R     025 MONBLK  003120R     025 M.RDCT= 000011          NUMQFL  003234R     025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R     025 M.REVS= 000003          NUMRIG  003231R     025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000          NUMWID  003226R     025
```

CMPTRT MACRO M1113 08-FEB-80 11:16 PAGE 46-3
SYMBOL TABLE

```
NUMZFL  003235R     025 RARGBL  003204R     025 R.OFF  = 000000          SY.CLR= 000040          T.LIT  = 000020
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.ERP= 000020          T.MON  = 000010
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR  = 000000          SY.NOU= 000100          T.NIL  = 000004
N.DESC= 000020          RARGLE  003216R     025 R.SEQ  = 000003          SY.OPN= 000002          T.NUM  = 002000
N.DEV = 000032          RARGLN  003210R     025 R.VAR  = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034          RARGPT  003206R     025 R6     =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DID = 000024          RARGSF  003214R     025 R7     =%000007          SY.SVC= 000004          T.SOP  = 000002
N.DVNM= 000032          RARGTY  003212R     025 SI     =%000005          SY.SVL= 000200          T.SPOP= 000040
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433           S.AB   = 177776          T.TMP  = 000200
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAL  = 100000
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R     025 S.BFHD= 000020          T.VAR  = 001000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R     025 S.BITS= 000002          UPPER  = 177400
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R     025 S.CHR1= 000003          UTNAME  000344R     023
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLE  007400R     025
N.PPN  = 000024         REENTR  003060R     025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R     025
N.UNIT= 000034          RESCT = ****** GX       SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
OBUFLN= 001000          RESCVT  003152R     025 SI.VAL= 177776           S.FNBW= 000017          WIDTH   000072R     025
OCHAN   005052R     025 RESERR= 104440           SKPSIZ  005134R     025 S.FNTY= 000004          WROUND  000020R     025
OMODE   000402R     023 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP0  005066R     025 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSFULL= 104403
OPTMP1  005070R     025 RESSFT  003146R     025 STKINC= 000774           S.LINK= 177774          WSNAME  000074R     025
OPTMP2  005072R     025 RESTYP  003144R     025 STOP   = 104420           S.NFEN= 000020          WSTIME  000132R     025
OPTMP3  005074R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP4  005076R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP5  005100R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP7  005104R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP8  005106R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSX   = ****** G
OPTMP9  005110R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          $RT11  = ****** G
OPTM10  005112R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .BOOLE= 000002
OPTM14  005026R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622          .CHARA= 000000
ORIGIN  000030R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436          .DEF.  = 000002
OUTPTR  000066R     025 RT11   = 000000          SYMWRN= 000054          TYPE   = 000166          .FLOAT= 000006
OVROPR= 000000          R.BITS= 000003           SYNERR= 104407          T.BOL  = 020000          .FSRCB= ****** G
PATCH   007344R     025 R.FIX  = 000001          SYSERR= 104402          T.CON  = 004000          .INTEG= 000004
PRECSN  000052R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100          .MOLUN  000404R     023
RANDAC  003064R     025 R.LN   = 000002          SY.BAD= 000001          T.FUN  = 010000          ...GBL= 000000
RANDY   003062R     025 R.LNUM= 000010           SY.BRK= 100000          T.LIST= 000400          ...TPC= 000020
RANERR= 104411          R.NEXT= 000006           SY.BS  = 00C010

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
CMPTRT  MACRO M1113  08-FEB-80 11:16  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000206      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12954 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:58
CMPTRT,DM1:[305,301]CMPTRT/-SP=APLRX1,APLDEF,CMPTRT,END
```

FILEID**COMAND

```
  CCCCCCCC      OOOOOO      MM        MM      AAAAAA      NN        NN    DDDDDDDD
  CCCCCCCC      OOOOOO      MM        MM      AAAAAA      NN        NN    DDDDDDDD
CC            OO      OO    MMMM    MMMM    AA    AA    NN        NN    DD      DD
CC            OO      OO    MMMM    MMMM    AA    AA    NN        NN    DD      DD
CC            OO      OO    MM  MM  MM    AA    AA    NNNN      NN    DD      DD
CC            OO      OO    MM  MM  MM    AA    AA    NNNN      NN    DD      DD
CC            OO      OO    MM      MM    AA    AA    NN  NN    NN    DD      DD
CC            OO      OO    MM      MM    AA    AA    NN  NN    NN    DD      DD
CC            OO      OO    MM      MM    AAAAAAAAAA    NN    NNNN    DD      DD
CC            OO      OO    MM      MM    AAAAAAAAAA    NN    NNNN    DD      DD
CC            OO      OO    MM      MM    AA    AA    NN        NN    DD      DD
CC            OO      OO    MM      MM    AA    AA    NN        NN    DD      DD
  CCCCCCCC      OOOOOO      MM      MM    AA    AA    NN        NN    DDDDDDDD
  CCCCCCCC      OOOOOO      MM      MM    AA    AA    NN        NN    DDDDDDDD

LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:18
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS     OPERATOR OVERLAYS
     5-  268    DEFINITIONS     REGISTERS
     6-  331    DEFINITIONS     MEMORY ORGANIZATION
     7-  377    DEFINITIONS     ORDERING PSECTS
     8-  426    DEFINITIONS     SI PARSE FRAMES
     9-  474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS     SI EXECUTION FRAMES
    11-  580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS     CHRTAB BITS
    12-  624    DEFINITIONS     SYSTEM STATUS BITS
    13-  640    DEFINITIONS     F-FRAMES
    14-  724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS     LIST STORAGE BLOCK
    18-  911    DEFINITIONS     INDEXING STACK FRAME
    19-  940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS     CHANNEL BLOCK
    22- 1085    DEFINITIONS             RSX DATA
    23- 1110    DEFINITIONS     NAME BLOCKS
    33- 1612    DEFINITIONS     MACROS
    35- 1905    DEFINITIONS     APL CHARACTERS
    36- 1967    DEFINITIONS     ERRORS
    44-    4    COMAND          COMMAND DISPATCHER
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 1


      1          000000                       RT11    =  0          ;NOT RT-11
      2          000000                       RSTS    =  0          ;NOT RSTS/E
      3          000115                       RSX     = 'M          ;RSX-11/M
      4
      5          000000                       EIS     =  0          ;NO EIS
      6          000000                       FIS     =  0          ;NO FIS
      7          000000                       FPP     =  0          ;NO FPP
      8
      9          000004                       FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                       DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:18  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  COMAND
                                .IDENT  \V02.03\

                                .SBTTL  COMAND           COMMAND DISPATCHER

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'COMAND' IS CALLED BY 'EXEC' WHEN A RIGHT PARENTHESIS
                                ;       IS FOUND AT THE START OF A LINE OF INPUT.  ALL COMMANDS
                                ;       ARE ENTERED IN THE COMMAND TABLE AS SYMBOLS, SO ALL THIS
                                ;       ROUTINE HAS TO DO IS TO LOOK UP THE SYMBOL AND CHECK TO
                                ;       SEE THAT IT REALLY IS A COMMAND.  THEN ALL THAT IS
                                ;       REQUIRED IS TO BRANCH TO THE PROPER COMMAND
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    COMAND          ;Only from 'EXEC'
                                ;
                                ; INPUTS:
                                ;       R0      - INPUT CHAR POINTER
                                ;       R1      - INPUT CHAR COUNT
                                ;
                                ; ROUTINES CALLED:
                                ;       CARVE   - Clean up the result Character data
                                ;                 for Execution Frames.
                                ;       PKILL   - Kill Parse Frames on SI
                                ;       SETBIG  - Get space for result of Execution Frame
                                ;
                                ;       and all the various command routines
                                ;
                                ; SIDE EFFECTS:
                                ;       NO REGISTERS ARE PRESERVED THROUGH THIS ROUTINE
                                ;--
000000                                  .PSECT  CODE

000000                          COMAND::
000000  010004                          MOV     R0,R4           ;COPY INPUT POINTER
000002  010103                          MOV     R1,R3           ;COPY INPUT COUNT
000004  032737  000400  004764'         BIT     #F$EXEC,FLAGS   ;EXECUTION FRAME?
000012  001416                          BEQ     13$             ;NOPE
000014                                  PUSH    <CHANEX,OCHAN,CHANEX+C.CCNT,CHANEX+C.CPTR>
                                                                ;SAVE OUTPUT POINTERS
000034  105037  000403'                 CLRB    TMODE           ;SET APL OUTPUT MODE (NECESSARY!)
000040                                  CALL    SETBIG          ;SET UP TO STORE OUTPUT
000044  062705  177772                  ADD     #SI.LEN,SI      ;AND CLEAR FRAME FROM SI STACK
000050  005000                  13$:    CLR     R0              ;NO CHR'S FOR NAME YET
000052  012701  003250'                 MOV     #CMDBUF,R1      ;POINT TO COMMAND BUFFER
000056                          1$:     GETC    3$              ;GET A CHARACTER
000056  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000060  002466                          BLT 3$                  ;NO, BRANCH TO 3$
000062  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000064  042702  177400                  BIC     #UPPER,R2       ;KILL GARBAGE BITS
000070  132762  000010  000000G         BITB    #C.IDEN, CHRTAB(R2);IDENTIFIER?
000076  001006                          BNE     2$              ;YES, GET THE COMMAND
000100  122702  000040                  CMPB    #CHRSPA,R2      ;ELSE WAS IT A SPACE?
000104  001764                          BEQ     1$              ;YES, JUST IGNORE IT

000106                          5$:     GETC    7$              ;GET A CHARACTER
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 44-1
COMAND           COMMAND DISPATCHER

        000106  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
        000110  002425                          BLT 7$                  ;NO, BRANCH TO 7$
        000112  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
     55 000114  042702  177400          2$:     BIC     #UPPER,R2       ;MAKE 8 BIT CHR
     56 000120  120227  000000G                 CMPB    R2,#CHRULA      ;GREATER THEN UNDERLINE 'A'
     57 000124  103405                          BLO     6$              ;NO
     58 000126  120227  000000G                 CMPB    R2,#CHRULZ      ;LESS THAN UNDERLINE 'Z'?
     59 000132  101002                          BHI     6$              ;NO
     60 000134  162702  177637G                 SUB     #CHRULA-CHR..A,R2;MAKE UPPERCASE LETTERS
     61 000140  132762  000010  000000G 6$:     BITB    #C.IDEN,CHRTAB(R2);IS IT AN IDENTIFIER?
     62 000146  001406                          BEQ     7$              ;NO, END OF COMMAND
     63 000150  110221                          MOVB    R2,(R1)+        ;SAVE IT
     64 000152  005200                          INC     R0              ;COUNT A CHR
     65 000154  020027  000010                  CMP     R0,#CMDMAX      ;BUFFER FULL?
     66 000160  101752                          BLOS    5$              ;NO
     67 000162  000425                          BR      3$              ;YES
     68
     69 000164                          7$:     PUSH    <R2,R3,R4>      ;SAVE THE LAST CHR AND POINTERS
     70 000172  012702  000001'                 MOV     #CMDNAM,R2      ;POINT TO TABLE OF COMMAND NAMES
     71
     72 000176  122200                  8$:     CMPB    (R2)+,R0        ;SAME LEN?
     73 000200  001011                          BNE     10$             ;NO
     74 000202  010203                          MOV     R2,R3           ;COPY POINTER
     75 000204  010004                          MOV     R0,R4           ;COPY COUNT
     76 000206  012701  003250'                 MOV     #CMDBUF,R1      ;POINT TO COMMAND JUST FOUND
     77 000212  122321                  9$:     CMPB    (R3)+,(R1)+     ;SAME?
     78 000214  001003                          BNE     10$             ;NO
     79 000216                                  SOB     R4,9$           ;TEST ALL CHR'S
     80 000222  000406                          BR      CMD1            ;MATCH!! WE FOUND IT!!
     81
     82 000224  116203  177777          10$:    MOVB    -1(R2),R3       ;GET COMMAND LENGTH
     83 000230  060302                          ADD     R3,R2           ;NEXT COMMAND
     84 000232  105722                          TSTB    (R2)+           ;MORE COMMANDS?
     85 000234  001360                          BNE     8$              ;YES!
     86 000236  104425                  3$:     COMERR                  ;COMMAND ERROR
     87
     88 000240  116200  177776          CMD1:   MOVB    -2(R2),R0       ;GET COMMAND NUMBER
     89 000244  013701  003314'                 MOV     SIFRAM,R1       ;POINT TO CURRENT FRAME
     90 000250                                  POP     <R4,R3,R2>      ;RESTORE INPUT DATA
     91 000256  005203                          INC     R3              ;BACKUP ONE CHAR
     92 000260  005304                          DEC     R4              ;FOR )WSID
     93 000262  112702  000040                  MOVB    #CHRSPA,R2      ;PUT A SPACE AS NEXT CHAR
     94 000266                                  PUSH    <F.LSTC(R1),SI> ;SAVE CURRENT FRAME POINTERS
     95 000274                                  CALL    @COMTAB(R0)     ;DO COMMAND
     96 000300  013701  003314'                 MOV     SIFRAM,R1       ;POINT TO CURRENT FRAME
     97 000304                                  POP     <SI,F.LSTC(R1)> ;RESTORE FRAME POINTERS
     98 000312  032737  000400  004764'         BIT     #F$EXEC,FLAGS   ;EXECUTION FRAME?
     99 000320  001441                          BEQ     12$             ;NOPE
    100 000322                                  CALL    PKILL           ;KILL ANY PARSE FRAMES CRE'TED
    101 000326  013700  003040'                 MOV     CHANEX+C.BUFF,R0;OF THE EXECUTE OP
    102 000332                                  SIFRAM  ,#T.VAL         ;Put Execute Buffer on SI (it's the result)
    103 000344  005037  003040'                 CLR     CHANEX+C.BUFF   ;NOW CLEAR BUFFER POINTER
    104 000350  005037  003060'                 CLR     CHANEX          ;FILE ISN'T OPEN!
    105 000354  013701  003044'                 MOV     CHANEX+C.CPTR,R1;GET CURRENT POINTER
    106 000360  161001                          SUB     (R0),R1         ;CALCULATE REAL LENGTH OF DATA
    107 000362  010160  000004                  MOV     R1,A.ALEN(R0)   ;REAL ALEN
    108 000366  010160  000006                  MOV     R1,A.DIM1(R0)   ;REAL DIM1
```

```
    109 000372                                      CALL    CARVE           ;KILL UNUSED PORTION
    110 000376                                      POP     <CHANEX+C.CPTR,CHANEX+C.CCNT,OCHAN,CHANEX>;RESTORE OUTPUT POINTERS
    111 000416  113737  000402' 000403'             MOVB    OMODE,TMODE     ;RESET OUTPUT MODE
    112 000424                              12$:    RETURN
    113
```

```
                                ;++
                                ;  THIS TABLE CONTAINS THE COMMAND NUMBER, LENGTH, AND CHARACTERS.
                                ;ALL ARE IN BYTES AND APPEAR AS:
                                ;
                                ;       NUMBER (OFFSET IN 'COMTAB')
                                ;       LENGTH (NUMBER OF CHARACTERS)
                                ;       CHAR #1
                                ;       CHAR #2
                                ;         .
                                ;         .
                                ;--

                                .MACRO  CMD     RTNPTR, NAME
                                    .NLIST
                                ;
                                ; The Pointer in the dispatch table
                                ;
                                    .PSECT  TABLES

                                TMPTAG= .                       ;Pointer to loc'n in table
                                    .WORD   RTNPTR
                                ;
                                ; The data in the Command Lookup Table
                                ;
                                    .PSECT  COMTB

                                    .BYTE   TMPTAG-COMTAB
                                    .NCHR   NAMELN,<NAME>
                                    .BYTE   NAMELN
                                    .IRPC   CHR,NAME
                                        .IF NB CHR
                                            .BYTE   ''CHR+40
                                        .ENDC
                                    .ENDR
                                    .LIST
                                .ENDM

                                ;
                                ; This is for the dispatch table entries
                                ;
000000                                  .PSECT  TABLES
000000                          COMTAB:

                                ;
                                ; This is for the list of commands
                                ;
000000                                  .PSECT  COMTB
        000001'                         .=      .+1
000001                          CMDNAM:
        000000'                         .=      .-1


000000                                  CMD     ASSFIL, ASSIGN                  ;ASSIGN
000010                                  CMD     CLEAR,  CLEAR                   ;CLEAR
000017                                  CMD     CLSFIL, CLOSE                   ;CLOSE
000026                                  CMD     CNTFIL, CONTINUE                ;CONTINUE
000040                                  CMD     COPYC,  COPY                    ;COPY
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 45-1
COMAND          COMMAND DISPATCHER

    172 000046                          CMD     CRTFIL, CREATE          ;CREATE
    173 000056                          CMD     DIGIT,  DIGITS          ;DIGITS
    174 000066                          CMD     DROPC,  DROP            ;DROP
    175 000074                          CMD     ERASEC, ERASE           ;ERASE
    176 000103                          CMD     FNS,    FNS             ;FNS
    177 000110                          CMD     FUZZY,  FUZZ            ;FUZZ
    178 000116                          CMD     GROUPC, GROUP           ;GROUP
    179 000125                          CMD     GRPC,   GRP             ;GRP
    180 000132                          CMD     GRPS,   GRPS            ;GRPS
    181 000140                          CMD     LIBC,   LIB             ;LIB
    182 000145                          CMD     LOAD,   LOAD            ;LOAD
    183 000153                          CMD     OFF,    OFF             ;OFF
    184 000160                          CMD     ORIGNN, ORIGIN          ;ORIGIN
    185 000170                          CMD     PCOPYC, PCOPY           ;PCOPY
    186 000177                          CMD     RENFIL, RENAME          ;RENAME
    190 000207                          CMD     SAVE,   SAVE            ;SAVE
    191 000215                          CMD     SILIS,  SI              ;SI
    192 000221                          CMD     SIV,    SIV             ;SIV
    193 000226                          CMD     NYIER,  SYMBOL          ;SYMBOL
    194 000236                          CMD     VARS,   VARS            ;VARS
    195 000244                          CMD     WIDTTH, WIDTH           ;WIDTH
    196 000253                          CMD     WSID,   WSID            ;WSID
    197
    198 000261                          .PSECT  COMTB
    199 000261      000                 .BYTE   0                       ;End of Commands
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 47
COMAND           COMMAND DISPATCHER

      1          000001                          .END
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 47-1
SYMBOL TABLE

AC0    =%000000          CHRFIL= 000177          COMTAB  000000R     002 DROPC = ****** GX         D.TRAN= 000004
AC1    =%000001          CHRFRO= 000135          COPYC = ****** GX       DYDCNT  003224R     025 D.UNON= 000006
AC2    =%000002          CHRLBR= 000073          CORPTR  005120R     025 D.AND = 000000          D.$MAX= 000000
AC3    =%000003          CHRLPR= 000072          CRTFIL= ****** GX       D.BXCT= 000007          D.$MIN= 000000
AC4    =%000004          CHRMNS= 000137          CS.ASS= 000200          D.BXIO= 000007          D.$MIL= 000000
AC5    =%000005          CHRMUL= 000075          CS.BOO= 000010          D.BXNL= 000007          D.$PLU= 000000
ALLINT  003274R     025  CHRNEG= 000100          CS.FLO= 000020          D.BXPP= 000007          D.$RES= 000000
ARYRET  003276R     025  CHRPDL= 000271          CS.IDN= 000040          D.BXPW= 000007          D.$SUB= 000000
ASCQUO= 000042           CHRPLU= 000055          CS.ONE= 000100          D.BXRL= 000007          D.1COM= 000001
ASSFIL= ****** GX        CHRPWR= 000120          CTEPS   000152R     025 D.CIRC= 000001          D.1EXP= 000002
A.ALEN= 000004           CHRQST= 000121          CTMAX   000212R     025 D.COMB= 000001          D.1ROT= 000003
A.ARG = 000040           CHRQUO= 000113          CTRMAX  000172R     025 D.COMP= 000001          EDICHR  005063R     025
A.ASB = 000200           CHRRBR= 000047          C.ALOC= 000162          D.CONT= 000006          EDICNT  005064R     025
A.BITS= 000003           CHRRPR= 000042          C.BLOK= 000064          D.CTLM= 000000          EDIFLG  005062R     025
A.CON = 000020           CHRSEM= 000074          C.BLSZ= 000140          D.DEAL= 000003          EDIT   = 000003
A.DIMS= 000002           CHRSMO= 000112          C.BUFF= 177760          D.DECD= 000002          EIS    = 000000
A.DIM1= 000006           CHRSPA= 000040          C.BUFL= 177762          D.DIV = 000000          ELESIZ  005132R     025
A.NUM = 000006           CHRTAB= ****** GX       C.CCNT= 177766          D.DMD = 000010          ENDCOR  000004R     025
A.PTR = 000000           CHRULA= ****** GX       C.CPTR= 177764          D.DQAD= 177777          EOF    = 104475
A.TMP = 000010           CHRULZ= ****** GX       C.CZ  = 001000          D.DROP= 000002          ERASEC= ****** GX
BADDIR= 104463           CHRUPA= 000131          C.DIG = 000040          D.ENCD= 000002          ERRMAX= 000166
BADERR= 104437           CHR.BS= 000010          C.FASW= 020000          D.EQ  = 000000          ERRNUM  004772R     025
BLZBUB  005116R     025  CHR.CC= 000003          C.FLEN= 000010          D.EXPD= 000002          ERRPTR  004770R     025
BRSELF  005046R     025  CHR.CR= 000015          C.FLSH= 004000          D.FGET= 000005          ERR0    004774R     025
CARVE = ****** GX        CHR.CZ= 000032          C.FRC = 000001          D.FMT = 000004          EXEBUF  005054R     025
CCNT    005072R     025  CHR.GT= 000046          C.IDEN= 000010          D.FPNT= 000005          E$CT    005056R     025
CHANEX  003060R     025  CHR.LF= 000012          C.NAME= 000102          D.FPUT= 000005          E$PT    005060R     025
CHAN00  000256R     025  CHR.LT= 000043          C.NOBF= 000400          D.GE  = 000000          FA.APD= 000100
CHAN01  000330R     025  CHR.SI= 000017          C.NUM = 000020          D.GT  = 000000          FA.CRE= 000010
CHAN02  000512R     025  CHR.SL= 000057          C.OP  = 000004          D.IBM = 000002          FA.DLK= 001000
CHAN03  000674R     025  CHR.SO= 000016          C.OPN = 010000          D.IND = 000002          FA.ENB= 100000
CHAN04  001056R     025  CHR.TB= 000011          C.RDLN= 177776          D.INNR= 000013          FA.EXC= 002000
CHAN05  001240R     025  CHR.UL= 000106          C.REWT= 002000          D.INTR= 000006          FA.EXT= 000004
CHAN06  001422R     025  CHR..A= 000141          C.RND = 000010          D.LE  = 000000          FA.NSP= 000100
CHAN07  001604R     025  CHR..E= 000145          C.SBIT= 177756          D.LOG = 000001          FA.NSR= 000200
CHAN08  001766R     025  CHR..S= 000163          C.SPFX= 000100          D.LT  = 000000          FA.POS= 010000
CHAN09  002150R     025  CHR..T= 000164          C.SPGL= 000200          D.MAX = 000000          FA.RD = 000001
CHAN10  002332R     025  CHR..U= 000165          C.SPLX= 000001          D.MEM = 000002          FA.RWD= 004000
CHAN11  002514R     025  CHR..Z= 000172          C.SPRS= 000002          D.MIN = 000000          FA.SEQ= 040000
CHAN12  002676R     025  CHR..0= 000060          C.SRC = 177774          D.MUL = 000000          FA.SHR= 000040
CHAN13  000022R     023  CHR..1= 000061          C.SRCP= 000024          D.NE  = 000000          FA.TMP= 000020
CHAN14  000204R     023  CH.AND= 000001          C.TT  = 000004          D.NN  = 000000          FA.WCK= 020000
CHNOPN= 104471           CLEAR = ****** GX       C.WIDC= 177770          D.NR  = 000000          FA.WRT= 000002
CHRAMD= 000051           CLSFIL= ****** GX       C.WIDR= 177772          D.OR  = 000000          FD.BLK= 000010
CHRBEL= 000007           CMDBUF  003250R     025 DEBUG = 000000          D.PLUS= 000000          FD.CCL= 000002
CHRBKR= 000133           CMDMAX= 000010          DEFERR= 104405          D.PSUB= 000006          FD.COM= 020000
CHRBOL= 000200           CMDNAM  000001R     010 DEVIL   005114R     025 D.PWR = 000001          FD.CR = 000002
CHRBOX= 000114           CMD1    000240R     020 DEVNTF= 104470          D.QQAD= 000000          FD.DIR= 000010
CHRCMA= 000054           CNTFIL= ****** GX       DIGIT = ****** GX       D.QUAD= 000000          FD.FTN= 000001
CHRCOL= 000076           CNTSIZ  005130R     025 DIGITS  000006R     025 D.RES = 000000          FD.F11= 040000
CHRDEL= 000107           COMAND  000000RG    020 DIMAX = 000020          D.RHO = 000000          FD.INS= 000010
CHRDIV= 000053           COMBIN  005076R     025 DIMBLK  003112R     025 D.ROT = 000003          FD.ISP= 002000
CHRDLR= 000176           COMCNT  005126R     025 DIMPTR  003114R     025 D.SCON= 000006          FD.MNT= 100000
CHRDLT= 000110           COMERR= 104425          DIMTYP  003116R     025 D.SDIF= 000006          FD.OSP= 004000
CHRDOT= 000056           COMPTR  005122R     025 DIRFUL= 104466          D.SUB = 000000          FD.PLC= 000004
CHREP = 000105           COMRTN  003156R     025 DOMERR= 104417          D.SUBS= 000006          FD.PRN= 000004
CHRERR= 104431           COMSKP  005124R     025 DPTERR= 104416          D.TAKE= 000002          FD.PSE= 010000
```

COMAND MACRO M1113 08-FEB-80 11:18 PAGE 47-2
SYMBOL TABLE

```
FD.RAH= 000001          F.BKVB= 000064          F.SPUN= 000074          LINC    005102R 025 M.IBM = 000003
FD.RAN= 000002          F.BR  = 000020          F.STBK= 000036          LINE    005016R 025 M.IDN = 000000
FD.REC= 000001          F.CHR = 000075          F.STK = 000022          LININC  005026R 025 M.INSP= 000006
FD.RWM= 000001          F.CNTG= 000034          F.STOP= 000004          LISIZE= 000020      M.IOTA= 000000
FD.SDI= 000020          F.CODE= 000010          F.STP = 000001          LITOP   005072R 025 M.LIT = 000205
FD.SQD= 000040          F.DFNB= 000046          F.SUSP= 000002          LNLNG = 104541      M.LN  = 000001
FD.TTY= 000004          F.DSPT= 000044          F.TRC = 000002          LOAD  = ****** GX   M.MMD = 000010
FD.WBH= 000002          F.DVNM= 000134          F.UNIT= 000136          LOCLCT  005036R 025 M.NEG = 000000
FF.CHR= 000005          F.EFBK= 000010          F.URBD= 000020          LOWCOR  000002R 025 M.NOT = 000000
FF.NV = 000003          F.EFN = 000050          F.VAL = 177776          LSTERP  004766R 025 M.NUB = 000006
FF.POE= 000002          F.END = 002000          F.VBN = 000064          LWIDTH  000070R 025 M.NUM = 000204
FF.RWD= 000001          F.EOBB= 000032          F.VBSZ= 000060          L.BITS= 000003      M.OUTR= 000012
FF.RWF= 000006          F.ERR = 000052          GNOME   003102R 025     L.CNT = 000002      M.PITM= 000000
FF.SPC= 000004          F.FACC= 000043          GROUPC= ****** GX       L.PRNT= 000001      M.PRDS= 000006
FHEADR  005012R 025     F.FFBY= 000014          GRPC  = ****** GX       L.PTR = 000000      M.QQAD= 000000
FHRET   005040R 025     F.FLAG= 000002          GRPS  = ****** GX       L.PTR1= 000004      M.QUAD= 000000
FILNTF= 104467          F.FLB = 000100          HASHC   005136R 025     L.SUBS= 000002      M.RAND= 000012
FIS   = 000000          F.FNAM= 000110          HASHTB  005140R 025     MACHIN= 000106      M.RAVL= 000000
FISCON= 046000          F.FNB = 000102          HSHMSK= 177601          MAXCHN= 000014      M.RCIP= 000000
FIXPTR  003074R 025     F.FNDF= 000010          HSHSIZ= 000200          MONBLK  003120R 025 M.RDCT= 000011
FLAGS   004764R 025     F.FTYP= 000116          ICHAN   005050R 025     MONCVT  003132R 025 M.REVS= 000003
FLCBUF  003250R 025     F.FUN = 000010          ILFNAM= 104464          MONLEN  003130R 025 M.ROLL= 000000
FLINE   005014R 025     F.FVER= 000120          ILLCHN= 104540          MONPTR  003122R 025 M.SATA= 000001
FLOEXP  003270R 025     F.HEAD= 000006          ILLSW = 104565          MONSFT  003126R 025 M.SCAN= 000011
FLONEG  003272R 025     F.HIBK= 000004          INDERR= 104410          MONTYP  003124R 025 M.SHAP= 000000
FLSWTB  000222R 025     F.LBL = 000020          INNTMP  003250R 025     MONVAL  003134R 025 M.SIGN= 000000
FLTLEN= 000010          F.LINE= 000012          INNTM2  005016R 025     M.ABS = 000000      M.SUBR= 000001
FNINSI  005044R 025     F.LOC = 000004          INPTR   000064R 025     M.BOL = 000200      M.SYM = 000203
FNS   = ****** GX       F.LSTC= 000004          INSNAM= 104462          M.BXAV= 000007      M.SYN = 000201
FORIGN  000032R 025     F.LUN = 000042          IOERR = 104477          M.BXCR= 000007      M.TRAN= 000012
FO.APD= 000106          F.MBCT= 000054          I.CNT = 177776          M.BXCT= 000007      M.$ABS= 000000
FO.MFY= 000002          F.MBC1= 000055          I.DIM = 177772          M.BXEX= 000007      M.$CEI= 000000
FO.RD = 000001          F.MBFG= 000056          I.LEN = 000012          M.BXFX= 000007      M.$FLR= 000000
FO.UPD= 000006          F.MON = 000002          I.OERR= 000062          M.BXIO= 000007      M.$IDN= 000000
FO.WRT= 000016          F.NIL = 000001          I.OLD = 177770          M.BXLC= 000003      M.$NEG= 000000
FPMODE  000042R 025     F.NRBD= 000024          I.PTR = 177774          M.BXNC= 000007      M.$SIG= 000000
FPP   = 000000          F.NREC= 000030          I.STR = 177766          M.BXNL= 000007      M.1RDT= 000011
FPPLEN= 000004          F.OPN = 000010          LABERR= 104406          M.BXPP= 000007      M.1REV= 000003
FREBLK  007400R 025     F.OVBS= 000030          LABLPT  005042R 025     M.BXPW= 000007      M.1SCN= 000011
FUZZ    000142R 025     F.PREV= 000000          LARGBL  003162R 025     M.BXRL= 000007      NAMBUF  003240R 025
FUZZY = ****** GX       F.PRNT= 100000          LARGCV  003176R 025     M.BXTS= 000007      NAMELN= 000004
FXPTR1  003076R 025     F.PROT= 010000          LARGFL  003160R 025     M.BXWA= 000007      NB.DEV= 000200
F$EVAL= 000100          F.RACC= 000016          LARGLE  003174R 025     M.CEI = 177777      NB.DIR= 000100
F$EVI = 000200          F.RATT= 000001          LARGLN  003166R 025     M.CHR = 000202      NB.NAM= 000004
F$EXEC= 000400          F.RCNM= 000034          LARGPT  003164R 025     M.CMT = 000206      NB.SD1= 000400
F$FUNC= 001000          F.RCTL= 000017          LARGSF  003172R 025     M.DOUT= 000006      NB.SD2= 001000
F.ACTL= 000076          F.READ= 004000          LARGTY  003170R 025     M.DQAD= 000006      NB.SNM= 000040
F.ALOC= 000040          F.REDF= 000040          LARGVA  003200R 025     M.ENCD= 000004      NB.STP= 000020
F.BBFS= 000062          F.RES = 000040          LBINC   005100R 025     M.EXE = 000000      NB.SVR= 000010
F.BDB = 000070          F.RET = 177772          LBUFF   003316R 025     M.EXP = 000001      NB.TYP= 000002
F.BGBC= 000057          F.RSIZ= 000002          LBUFLN= 000621          M.FACT= 000001      NB.VER= 000001
F.BITS= 177774          F.RTYP= 000000          LCOUNT  005106R 025     M.FIL = 000177      ND.DEV= 000200
F.BKDN= 000026          F.R2  = 000014          LENERR= 104412          M.FLR = 177777      ND.EX1= 000002
F.BKDS= 000020          F.R3  = 000016          LENTAB  000054R 025     M.FMT = 000004      ND.NAM= 000004
F.BKEF= 000050          F.R4  = 000020          LEXPTR  003100R 025     M.GDUP= 000003      ND.PPN= 000100
F.BKP1= 000051          F.SEQN= 000100          LFTOP   005066R 025     M.GDWN= 000003      ND.$EX= 000000
F.BKST= 000024          F.SPDV= 000072          LIBC  = ****** GX       M.GETI= 000006      ND.$E$= 000020
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 47-3
SYMBOL TABLE

ND.$NM= 000000          OPTMP2  005072R     025 RLTBAS  005066R     025 SYMTAB  005352R     025 T.CON = 004000
ND.$N$= 000040          OPTMP3  005074R     025 ROTINC  005070R     025 SYMTBF= 104404          T.DIM = 000100
ND.$PN= 001400          OPTMP4  005076R     025 ROUND   000010R     025 SYMTOP  005350R     025 T.FUN = 010000
NEGSGN  005063R     025 OPTMP5  005100R     025 ROVER   003312R     025 SYMTP   006272R     025 T.LIST= 000400
NINPT1  003302R     025 OPTMP6  005102R     025 RSTS  = 000000          SYMWRN= 000054          T.LIT = 000020
NINPT2  003300R     025 OPTMP7  005104R     025 RSX   = 000115          SYNERR= 104407          T.MON = 000010
NINSTK  003304R     025 OPTMP8  005106R     025 RTOP    005070R     025 SYSER = ****** GX       T.NIL = 000004
NINTMP  003310R     025 OPTMP9  005110R     025 RT11  = 000000          SYSERR= 104402          T.NUM = 002000
NINTYP  003306R     025 OPTM10  005112R     025 R.BITS= 000003          SYSTAT  000000R     024 T.OPFN= 000001
NOTAVL= 104472          OPTM14  005026R     025 R.FIX = 000001          SY.BAD= 000001          T.OTHR= 040000
NOTOPN= 104473          ORIGIN  000030R     025 R.LCT = 000002          SY.BRK= 100000          T.SOP = 000002
NPRT1   004762R     025 ORIGNN= ****** GX       R.LN  = 000002          SY.BS = 000010          T.SPOP= 000040
NULL    007364R     025 OUTPTR  000066R     025 R.LNUM= 000010          SY.CLR= 000040          T.TMP = 000200
NUMCHN= 000020          OVROPR= 000000          R.NEXT= 000006          SY.ERP= 000020          T.VAL = 100000
NUMDFL  003230R     025 PATCH   007344R     025 R.OFF = 000000          SY.NOU= 000100          T.VAR = 001000
NUMEFL  003233R     025 PCOPYC= ****** GX       R.PREV= 000004          SY.OPN= 000002          UPPER = 177400
NUMESN  003236R     025 PKILL = ****** GX       R.PTR = 000000          SY.PMT= 001000          UTNAME  000344R     023
NUMFLG  003230R     025 PRECSN  000052R     025 R.SEQ = 000003          SY.SRT= 000400          VALERR= 104413
NUMLEN  000062R     025 RANDAC  003064R     025 R.VAR = 000002          SY.SVC= 000004          VARBLE  007400R     025
NUMNEG  003232R     025 RANDY   003062R     025 R6    =%000006          SY.SVL= 000200          VARBLS  000000R     025
NUMQFL  003234R     025 RANERR= 104411          R7    =%000007          S.AB  = 177776          VARS  = ****** GX
NUMRIG  003231R     025 RARGBL  003204R     025 SAVE  = ****** GX       S.ASB = 100000          VERNUM  000000R     025
NUMWID  003226R     025 RARGCV  003220R     025 SETBIG= ****** GX       S.BFHD= 000020          VRSION= 000002
NUMZFL  003235R     025 RARGFL  003202R     025 SI    =%000005          S.BITS= 000002          WIDTH   000072R     025
NYIER = ****** GX       RARGLE  003216R     025 SIDAMN= 104433          S.CHR1= 000003          WIDTTH= ****** GX
NYIERR= 104432          RARGLN  003210R     025 SIFRAM  003314R     025 S.FATT= 000016          WROUND  000020R     025
N.BLSZ= 000036          RARGPT  003206R     025 SILIS = ****** GX       S.FDB = 000140          WSDATE  000136R     025
N.DESC= 000020          RARGSF  003214R     025 SISTAK  006350R     025 S.FLB = 040000          WSFULL= 104403
N.DEV = 000032          RARGTY  003212R     025 SITOP   006346R     025 S.FNAM= 000006          WSID  = ****** GX
N.DEVN= 000034          RARGVA  003222R     025 SITP    007144R     025 S.FNB = 000036          WSNAME  000074R     025
N.DID = 000024          RBASE   005074R     025 SIV   = ****** GX       S.FNBW= 000017          WSTIME  000132R     025
N.DVNM= 000032          RCOUNR  005112R     025 SIWARN= 000200          S.FNTY= 000004          $FIS  = ****** G
N.EXT = 000014          RCOUNT  005110R     025 SIZTAB  000044R     025 S.FTYP= 000002          $FPP  = ****** G
N.FID = 000000          REDCNT  005070R     025 SI.BIT= 177774          S.LEN = 000002          $FPPLN= ****** G
N.FNAM= 000006          REDIGN  005066R     025 SI.LEN= 177772          S.LINK= 177774          $MACHN= ****** G
N.FTYP= 000014          REDOFF  005072R     025 SI.RET= 177772          S.NFEN= 000020          $RSTS = ****** G
N.FVER= 000016          REENTR  003060R     025 SI.VAL= 177776          S.TYPE= 140000          $RSX  = ****** G
N.NAME= 000006          RENFIL= ****** GX       SKPSIZ  005134R     025 S.VAL = 000000          $RT11 = ****** G
N.NEXT= 000022          RESBIT  003142R     025 SPHOLD  000004R     024 TKDIMS  005072R     025 .BOOLE= 000002
N.PPN = 000024          RESCVT  003152R     025 SPINIT  000002R     024 TKFRAM  005066R     025 .CHARA= 000000
N.STAT= 000020          RESERR= 104440          STKINC= 000774          TKLEFT  005070R     025 .DEF. = 000002
N.UNIT= 000034          RESLEN  003150R     025 STOP  = 104420          TMODE   000403R     023 .FLOAT= 000006
OBUFLN= 001000          RESPTR  003140R     025 SUBCNT  003104R     025 TMPTAG= 000064R     002 .FSRCB= ****** G
OCHAN   005052R     025 RESSFT  003146R     025 SUBDCT  003106R     025 TTYBUF  004140R     025 .INTEG= 000004
OFF   = ****** GX       RESTYP  003144R     025 SUBPTR  003110R     025 TTYLEN= 000622          .MOLUN  000404R     023
OMODE   000402R     023 RESULT  003136R     025 SYMB1   005344R     025 TURKEY= 104436          ...GBL= 000000
OPTMP0  005066R     025 RESVAL  003154R     025 SYMF    005340R     025 TYPE  = 000166          ...TPC= 000020
OPTMP1  005070R     025 RINC    005104R     025 SYMPT1  005346R     025 T.BOL = 020000

. ABS.  000000      000
        000000      001
TABLES  000066      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
```

```
COMAND  MACRO M1113  08-FEB-80 11:18  PAGE 47-4
SYMBOL TABLE

COMTB   000262      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000426      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13344 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:20
COMAND,DM1:[305,301]COMAND/-SP=APLRX1,APLDEF,COMAND,END
```

```
**FILE**ID**COMRT
```

```
  CCCCCCCC      OOOOOO      MM      MM    RRRRRRRR      TTTTTTTTTT
  CCCCCCCC      OOOOOO      MM      MM    RRRRRRRR      TTTTTTTTTT
CC            OO      OO    MMMM  MMMM    RR      RR        TT
CC            OO      OO    MMMM  MMMM    RR      RR        TT
CC            OO      OO    MM  MM  MM    RR      RR        TT
CC            OO      OO    MM  MM  MM    RR      RR        TT
CC            OO      OO    MM      MM    RRRRRRRR          TT
CC            OO      OO    MM      MM    RRRRRRRR          TT
CC            OO      OO    MM      MM    RR  RR            TT
CC            OO      OO    MM      MM    RR  RR            TT
CC            OO      OO    MM      MM    RR    RR          TT          ....
CC            OO      OO    MM      MM    RR    RR          TT          ....
  CCCCCCCC      OOOOOO      MM      MM    RR      RR        TT          ....
  CCCCCCCC      OOOOOO      MM      MM    RR      RR        TT          ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                  TT
LL            SS                  TT
LL            SS                  TT
LL            SS                  TT
LL              SSSSSS            TT
LL              SSSSSS            TT
LL                    SS          TT
LL                    SS          TT
LL                    SS          TT
LL                    SS          TT
LLLLLLLLLL    SSSSSSSS            TT
LLLLLLLLLL    SSSSSSSS            TT
```

.MAIN. MACRO M1113 08-FEB-80 11:20
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    COMPAR                  COMPARE ROUTINES
```

```
COMRT   MACRO M1113  08-FEB-80 11:20  PAGE 1


      1        000000                  RT11    =  0            ;NOT RT-11
      2        000000                  RSTS    =  0            ;NOT RSTS/E
      3        000115                  RSX     = 'M            ;RSX-11/M
      4
      5        000000                  EIS     =  0            ;NO EIS
      6        000000                  FIS     =  0            ;NO FIS
      7        000000                  FPP     =  0            ;NO FPP
      8
      9        000004                  FPPLEN  =  4.           ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
COMRT   MACRO M1113  08-FEB-80 11:20  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:20  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  COMRT
                                .IDENT  \V02.03\

                                .SBTTL  COMPAR                  COMPARE ROUTINES

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       These Routines are used to compare arguments by
                                ;       the Scalar Operators, Grade Up/Down, Epsilon, Iota, Etc...
                                ;       They take Args in the Standard Places and (Non-Destructively)
                                ;       Return with the appropriate Condition Codes Set
                                ;--
000000                                  .PSECT  CODE

000000                          CMP.F:: STF     AC2,AC3         ;Copy the Right argument
000006                                  ABSF    AC3             ;Get its magnitude
000014                                  STF     AC1,AC0         ;Copy the Left argument
000022                                  ABSF    AC0             ;Get its magnitude
000030                                  CMPF    AC0,AC3         ;Compare the magnitudes of the arguments
000036                                  CFCC                    ;And use the large one
000042  003403                          BLE     1$              ;Left magnitude greater than the Right?
000044                                  STF     AC0,AC3         ;Yes, Save the Left magnitude
000052                          1$:     STF     AC1,AC0         ;Get the Left argument
000060                                  SUBF    AC2,AC0         ;  L - R
000066                                  ABSF    AC0             ;  | L - R
000074                                  MULF    FUZZ,AC3        ;  FUZZ x ( | L ) .ce ( | R )
000104                                  SUBF    AC3,AC0         ;  ( | L - R ) - FUZZ x ( | L ) .ce ( | R )
000112                                  CFCC                    ;Result > 0 ?
000116  003003                          BGT     CMP..F          ;Yes, Implies arguments not equal
000120  000257                          CCC                     ;No, Implies arguments fuzzy equal
000122  000264                          SEZ                     ;Set the zero bit
000124                                  RETURN

000126                          CMP..F: CMPF    AC1,AC2         ;Not equal, So what are they?
000134                                  CFCC                    ;Set the condition codes
000140                                  RETURN

000142  023737  003200' 003222' CMP.I:: CMP      LARGVA,RARGVA   ;COMPARE, INTEGER LEVEL
000150                                  RETURN                  ;FINISHED

000152                          CMP.B::
000152  123737  003200' 003222'         CMPB    LARGVA,RARGVA   ;COMPARE, BOOLEAN
000160                                  RETURN                  ;FINISHED

000162                          CMP.C::
000162  023737  003200' 003222'         CMP     LARGVA,RARGVA   ;COMPARE CHARACTERS AS WORDS (UNSIGNED)
000170                                  RETURN                  ;FINISHED
```

```
OMRT   MACRO M1113  08-FEB-80 11:20  PAGE 45
OMPAR                   COMPARE ROUTINES

     49 000000                                  .PSECT  TABLES
     50                                         .EVEN
     51
     52 000000                                  COMCT::
     53 000000  000162' 000152' 00014 '         .WORD   CMP.C,  CMP.B,  CMP.I,  CMP.F
        000006  000000'
     54 000010                                  ECOMCT::
     55 00001C  000162' 000152' 000142'         .WORD   CMP.C,  CMP.B,  CMP.I,  CMP..F
        000016  000126'
```

```
COMRT   MACRO M1113  08-FEB-80 11:20  PAGE 47
COMPAR                  COMPARE ROUTINES

      1         000007                          .END
```

COMRT MACRO M1113 08-FEB-80 11:20 PAGE 47-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073          CS.ONE= 000100          D.COMB= 000001          D.1ROT= 000003
AC1    =%000001          CHRLPR= 000072          CTEPS   000152R 025     D.COMP= 000001          ECOMCT  000010RG  002
AC2    =%000002          CHRMNS= 000137          CTMAX   000212R 025     D.CONT= 000006          EDICHR  005063R  025
AC3    =%000003          CHRMUL= 000075          CTRMAX  000172R 025     D.CTLM= 000000          EDICNT  005064R  025
AC4    =%000004          CHRNEG= 000100          C.ALOC= 000162          D.DEAL= 000003          EDIFLG  005062R  025
AC5    =%000005          CHRPDL= 000271          C.BLOK= 000064          D.DECD= 000002          EDIT   = 000003
ALLINT  003274R  025      CHRPLU= 000055          C.BLSZ= 000140          D.DIV = 000000          EIS    = 000000
ARYRET  003276R  025      CHRPWR= 000120          C.BUFF= 177760          D.DMD = 000010          ELESIZ  005132R  025
ASCQUO= 000042            CHRQST= 000121          C.BUFL= 177762          D.DQAD= 177777          ENDCOR  000004R  025
A.ALEN= 000004            CHRQUO= 000113          C.CCNT= 177766          D.DROP= 000002          EOF    = 104475
A.ARG = 000040            CHRRBR= 000047          C.CPTR= 177764          D.ENCD= 000002          ERRMAX= 000166
A.ASB = 000200            CHRRPR= 000042          C.CZ  = 001000          D.EQ  = 000000          ERRNUM  004772R  025
A.BITS= 000003            CHRSEM= 000074          C.DIG = 000040          D.EXPD= 000002          ERRPTR  004770R  025
A.CON = 000020            CHRSMO= 000112          C.FASW= 020000          D.FGET= 000005          ERRO    004774R  025
A.DIMS= 000002            CHRSPA= 000040          C.FLEN= 000010          D.FMT = 000004          EXEBUF  005054R  025
A.DIM1= 000006            CHRUPA= 000131          C.FLSH= 004000          D.FPNT= 000005          E$CT    005056R  025
A.NUM = 000006            CHR.BS= 000010          C.FRC = 000001          D.FPUT= 000005          E$PT    005060R  025
A.PTR = 000000            CHR.CC= 000003          C.IDEN= 000010          D.GE  = 000000          FA.APD= 000100
A.TMP = 000010            CHR.CR= 000015          C.NAME= 000102          D.GT  = 000000          FA.CRE= 000010
BADDIR= 104463            CHR.CZ= 000032          C.NOBF= 000400          D.IBM = 000002          FA.DLK= 001000
BADERR= 104437            CHR.GT= 000046          C.NUM = 000020          D.IND = 000002          FA.ENB= 100000
BLZBUB  005116R  025      CHR.LF= 000012          C.OP  = 000004          D.INNR= 000013          FA.EXC= 002000
BRSELF  005046R  025      CHR.LT= 000043          C.OPN = 010000          D.INTR= 000006          FA.EXT= 000004
CCNT    005072R  025      CHR.SI= 000017          C.RDLN= 177776          D.LE  = 000000          FA.NSP= 000100
CHANEX  003060R  025      CHR.SL= 000057          C.REWT= 002000          D.LOG = 000001          FA.NSR= 000200
CHAN00  000256R  025      CHR.SO= 000016          C.RND = 000010          D.LT  = 000000          FA.POS= 010000
CHAN01  000330R  025      CHR.TB= 000011          C.SBIT= 177756          D.MAX = 000000          FA.RD = 000001
CHAN02  000512R  025      CHR.UL= 000106          C.SPFX= 000100          D.MEM = 000002          FA.RWD= 004000
CHAN03  000674R  025      CHR..A= 000141          C.SPGL= 000200          D.MIN = 000000          FA.SEQ= 040000
CHAN04  001056R  025      CHR..E= 000145          C.SPLX= 000001          D.MUL = 000000          FA.SHR= 000040
CHAN05  001240R  025      CHR..S= 000163          C.SPRS= 000002          D.NE  = 000000          FA.TMP= 000020
CHAN06  001422R  025      CHR..T= 000164          C.SRC = 177774          D.NN  = 000000          FA.WCK= 020000
CHAN07  001604R  025      CHR..U= 000165          C.SRCP= 000024          D.NR  = 000000          FA.WRT= 000002
CHAN08  001766R  025      CHR..Z= 000172          C.TT  = 000004          D.OR  = 000000          FD.BLK= 000010
CHAN09  002150R  025      CHR..0= 000060          C.WIDC= 177770          D.PLUS= 000000          FD.CCL= 000002
CHAN10  002332R  025      CHR..1= 000061          C.WIDR= 177772          D.PSUB= 000006          FD.COM= 020000
CHAN11  002514R  025      CH.AND= 000001          DEBUG = 000000          D.PWR = 000001          FD.CR = 000002
CHAN12  002676R  025      CMDBUF  003250R  025    DEFERR= 104405          D.QQAD= 000000          FD.DIR= 000010
CHAN13  000022R  023      CMDMAX= 000010          DEVIL   005114R  025    D.QUAD= 000000          FD.FTN= 000001
CHAN14  000204R  023      CMP.B   000152RG  020   DEVNTF= 104470          D.RES = 000000          FD.F11= 040000
CHNOPN= 104471            CMP.C   000162RG  020   DIGITS  000006R  025    D.RHO = 000000          FD.INS= 000010
CHRAND= 000051            CMP.F   000000RG  020   DIMAX = 000020          D.ROT = 000003          FD.ISP= 002000
CHRBEL= 000007            CMP.I   000142RG  020   DIMBLK  003112R  025    D.SCON= 000006          FD.MNT= 100000
CHRBKR= 000133            CMP..F  000126R  020    DIMPTR  003114R  025    D.SDIF= 000006          FD.OSP= 004000
CHRBOL= 000200            CNTSIZ  005130R  025    DIMTYP  003116R  025    D.SUB = 000000          FD.PLC= 000004
CHRBOX= 000114            COMBIN  005076R  025    DIRFUL= 104466          D.SUBS= 000006          FD.PRN= 000004
CHRCMA= 000054            COMCNT  005126R  025    DOMERR= 104417          D.TAKE= 000002          FD.PSE= 010000
CHRCOL= 000076            COMCT   000000RG  002   DPTERR= 104416          D.TRAN= 000004          FD.RAH= 000001
CHRDEL= 000107            COMERR= 104425          DYDCNT  003224R  025    D.UNON= 000006          FD.RAN= 000002
CHRDIV= 000053            COMPTR  005122R  025    D.AND = 000000          D.$MAX= 000000          FD.REC= 000001
CHRDLR= 000176            COMRTN  003156R  025    D.BXCT= 000007          D.$MIN= 000000          FD.RWM= 000001
CHRDLT= 000110            COMSKP  005124R  025    D.BXIO= 000007          D.$MUL= 000000          FD.SDI= 000020
CHRDOT= 000056            CORPTR  005120R  025    D.BXNL= 000007          D.$PLU= 000000          FD.SQD= 000040
CHREP = 000105            CS.ASS= 000200          D.BXPP= 000007          D.$RES= 000000          FD.TTY= 000004
CHRERR= 104431            CS.BOO= 000010          D.BXPW= 000007          D.$SUB= 000000          FD.WBH= 000002
CHRFIL= 000177            CS.FLO= 000020          D.BXRL= 000007          D.1COM= 000001          FF.CHR= 000005
CHRFRO= 000135            CS.IDN= 000040          D.CIRC= 000001          D.1EXP= 000002          FF.NV = 000003
```

COMRT MACRO M1113 08-FEB-80 11:20 PAGE 47-2
SYMBOL TABLE

```
FF.POE= 000002          F.ERR = 000052          GNOME   003102R 025 MACHIN= 000106          M.RAVL= 000000
FF.RWD= 000001          F.FACC= 000043          HASHC   005136R 025 MAXCHN= 000014          M.RCIP= 000000
FF.RWF= 000006          F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011
FF.SPC= 000004          F.FLAG= 000002          HSHMSK= 177601      MONCVT  003132R 025 M.REVS= 000003
FHEADR  005012R 025 F.FLB = 000100              HSHSIZ= 000200      MONLEN  003130R 025 M.ROLL= 000000
FHRET   005040R 025 F.FNAM= 000110              ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025 F.FUN = 000010              INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025 F.FVER= 000120              INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025 F.HEAD= 000006              INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025 F.HIBK= 000004              INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025 F.LBL = 000020              INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025 F.LINE= 000012              IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025 F.LOC = 000004              I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025 F.LUN = 000042              I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025 F.MBCT= 000054              I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025 F.NREC= 000030              LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025 F.PREV= 000000              LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025 F.PRNT= 100000              LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025 F.PROT= 010000              LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025 M.GDWN= 000003      ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025 M.IBM = 000003      ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
```

```
COMRT   MACRO M1113  08-FEB-80 11:20  PAGE 47-3
SYMBOL TABLE

NUMLEN  000062R     025 RANDAC  003064R     025 R.OFF  = 000000          SY.NOU= 000100          T.OPFN= 000001
NUMNEG  003232R     025 RANDY   003062R     025 R.PREV= 000004          SY.OPN= 000002          T.OTHR= 040000
NUMQFL  003234R     025 RANERR= 104411          R.PTR  = 000000          SY.PMT= 001000          T.SOP = 000002
NUMRIG  003231R     025 RARGBL  003204R     025 R.SEQ  = 000003          SY.SRT= 000400          T.SPOP= 000040
NUMWID  003226R     025 RARGCV  003220R     025 R.VAR  = 000002          SY.SVC= 000004          T.TMP = 000200
NUMZFL  003235R     025 RARGFL  003202R     025 R6     =%000006          SY.SVL= 000200          T.VAL = 100000
NYIERR= 104432          RARGLE  003216R     025 R7     =%000007          S.AB   = 177776          T.VAR = 001000
N.BLSZ= 000036          RARGLN  003210R     025 SI     =%000005          S.ASB  = 100000          UPPER = 177400
N.DESC= 000020          RARGPT  003206R     025 SIDAMN= 104433           S.BFHD= 000020          UTNAME  000344R     023
N.DEV = 000032          RARGSF  003214R     025 SIFRAM  003314R     025 S.BITS= 000002          VALERR= 104413
N.DEVN= 000034          RARGTY  003212R     025 SISTAK  006350R     025 S.CHR1= 000003          VARBLE  007400R     025
N.DID = 000024          RARGVA  003222R     025 SITOP   006346R     025 S.FATT= 000016          VARBLS  000000R     025
N.DVNM= 000032          RBASE   005074R     025 SITP    007144R     025 S.FDB  = 000140          VERNUM  000000R     025
N.EXT = 000014          RCOUNR  005112R     025 SIWARN= 000200           S.FLB  = 040000          VRSION= 000002
N.FID = 000000          RCOUNT  005110R     025 SIZTAB  000044R     025 S.FNAM= 000006          WIDTH   000072R     025
N.FNAM= 000006          REDCNT  005070R     025 SI.BIT= 177774           S.FNB  = 000036          WROUND  000020R     025
N.FTYP= 000014          REDIGN  005066R     025 SI.LEN= 177772           S.FNBW= 000017          WSDATE  000136R     025
N.FVER= 000016          REDOFF  005072R     025 SI.RET= 177772           S.FNTY= 000004          WSFULL= 104403
N.NAME= 000006          REENTR  000060R     025 SI.VAL= 177776           S.FTYP= 000002          WSNAME  000074R     025
N.NEXT= 000022          RESBIT  000042R     025 SKPSIZ  005134R     025 S.LEN  = 000002          WSTIME  000132R     025
N.PPN  = 000024         RESCVT  003152R     025 SPHOLD  000004R     024 S.LINK= 177774          $ABSF  = ****** GX
N.STAT= 000020          RESERR= 104440          SPINIT  000002R     024 S.NFEN= 000020          $CFCC  = ****** GX
N.UNIT= 000034          RESLEN  003150R     025 STKINC= 000774           S.TYPE= 140000          $CMPF  = ****** GX
OBUFLN= 001000          RESPTR  003140R     025 STOP   = 104420          S.VAL  = 000000          $FIS   = ****** G
OCHAN   005052R     025 RESSFT  003146R     025 SUBCNT  003104R     025 TKDIMS  005072R     025 $FPP   = ****** G
OMODE   000402R     023 RESTYP  003144R     025 SUBDCT  003106R     025 TKFRAM  005066R     025 $FPPLN= ****** G
OPTMP0  005066R     025 RESULT  003136R     025 SUBPTR  003110R     025 TKLEFT  005070R     025 $MACHN= ****** G
OPTMP1  005070R     025 RESVAL  003154R     025 SYMB1   005344R     025 TMODE   000403R     023 $MULF  = ****** GX
OPTMP2  005072R     025 RINC    005104R     025 SYMF    005340R     025 TMPTAG= 000166          $RSTS  = ****** G
OPTMP3  005074R     025 RLTBAS  005066R     025 SYMPT1  005346R     025 TTYBUF  004140R     025 $RSX   = ****** G
OPTMP4  005076R     025 ROTINC  005070R     025 SYMTAB  005352R     025 TTYLEN= 000622          $RT11  = ****** G
OPTMP5  005100R     025 ROUND   000010R     025 SYMTBF= 104404          TURKEY= 104436          $STF   = ****** GX
OPTMP6  005102R     025 ROVER   003312R     025 SYMTOP  005350R     025 TYPE   = 000166          $SUBF  = ****** GX
OPTMP7  005104R     025 RSTS   = 000000         SYMTP   006272R     025 T.BOL  = 020000          .BOOLE= 000002
OPTMP8  005106R     025 RSX    = 000115         SYMWRN= 000054           T.CON  = 004000          .CHARA= 000000
OPTMP9  005110R     025 RTOP    005070R     025 SYNERR= 104407           T.DIM  = 000100          .DEF.  = 000002
OPTM10  005112R     025 RT11   = 000000         SYSERR= 104402           T.FUN  = 010000          .FLOAT= 000006
OPTM14  005026R     025 R.BITS= 000003          SYSTAT  000000R     024 T.LIST= 000400          .FSRCB= ****** G
ORIGIN  000030R     025 R.FIX  = 000001          SY.BAD= 000001           T.LIT  = 000020          .INTEG= 000004
OUTPTR  000066R     025 R.LCT  = 000002          SY.BRK= 100000           T.MON  = 000010          .MOLUN  000404R     023
OVROPR= 000000          R.LN   = 000002          SY.BS  = 000010          T.NIL  = 000004          ...GBL= 000000
PATCH   007344R     025 R.LNUM= 000010          SY.CLR= 000040           T.NUM  = 002000          ...TPC= 000020
PRECSN  000052R     025 R.NEXT= 000006          SY.ERP= 000020

. ABS.  000000      000
        000000      001
TABLES  000020      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
```

```
COMRT   MACRO M1113  08-FEB-80 11:20  PAGE 47-4
SYMBO'  TABLE

TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000172      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12954 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:47
COMRT,DM1:[305,301]COMRT/-SP=APLRX1,APLDEF,COMRT,END
```

FILEID**CONVER

```
  CCCCCCCC      000000      NN      NN   VV      VV   EEEEEEEEEE   RRRRRRRR
  CCCCCCCC      000000      NN      NN   VV      VV   EEEEEEEEEE   RRRRRRRR
CC            OO      OO   NN      NN   VV      VV   EE           RR      RR
CC            OO      OO   NN      NN   VV      VV   EE           RR      RR
CC            OO      OO   NNNN    NN   VV      VV   EE           RR      RR
CC            OO      OO   NNNN    NN   VV      VV   EE           RR      RR
CC            OO      OO   NN  NN  NN   VV      VV   EEEEEEEE     RRRRRRRR
CC            OO      OO   NN  NN  NN   VV      VV   EEEEEEEE     RRRRRRRR
CC            OO      OO   NN    NNNN    VV      VV   EE           RR  RR
CC            OO      OO   NN    NNNN    VV      VV   EE           RR  RR
CC            OO      OO   NN      NN      VV  VV     EE           RR    RR        ....
CC            OO      OO   NN      NN      VV  VV     EE           RR    RR        ....
  CCCCCCCC      000000      NN      NN        VV       EEEEEEEEEE   RR      RR      ....
  CCCCCCCC      000000      NN      NN        VV       EEEEEEEEEE   RR      RR      ....

LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:22
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF  THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    CONVERT          CHANGE TYPE OF DATA IN SI FRAME
```

```
CONVER  MACRO M1113  08-FEB-80 11:22  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CONVER  MACRO M1113  08-FEB-80 11:22  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CONVER
                                .IDENT  \V02.03\

                                .SBTTL  CONVERT         CHANGE TYPE OF DATA IN SI FRAME

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is passed a pointer to an SI frame in R2
                                ;       and a type description in R1. GETSM3 is called to create
                                ;       a (new) item on the SI. The type is converted from the
                                ;       old type to the new type if possible, otherwise
                                ;       a DOMAIN ERROR occurs.
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     SI,R2           ;point to val on SI to convert
                                ;       MOV     #<New Type>,R1  ;type to convert to
                                ;       CALL    CONVERT
                                ;
                                ; INPUTS:
                                ;       R1      - Type of result wanted: .CHARACTER, .BOOLEAN,
                                ;                 .INTEGER, .FLOAT
                                ;       R2      - Pointer to SI frame for array block to be converted
                                ;
                                ; OUTPUTS:
                                ;       R0      - Points to converted array storage block.
                                ;       R1      - <Preserved>
                                ;       R2      - <Preserved>
                                ;
                                ; ROUTINES CALLED:
                                ;       GETSM3  - Get a new block for result
                                ;       SM.B    - Store result as Boolean
                                ;       SM.I    - Store result as Integer
                                ;--
000000                          .PSECT  CODE
```

B	1		J	5		E	10		M	14					
C	1		K	5		F	10		N	14					
D	1		L	5		G	10		B	15					
E	1		M	5		H	10		C	15					
F	1		N	5		I	10	CMDCHN	D	15					
G	1		B	6		J	10		E	15					
H	1		C	6		K	10		F	15					
I	1	BINSCH	D	6		L	10		G	15					
J	1		E	6		M	10		H	15					
K	1		F	6		N	10		I	15	COMRT				
L	1		G	6		B	11		J	15					
M	1		H	6		C	11		K	15					
N	1		I	6		D	11		L	15					
B	2		J	6		E	11		M	15					
C	2		K	6		F	11		N	15					
D	2		L	6		G	11	CMDNBR	B	16					
E	2		M	6		H	11		C	16					
F	2		N	6	CKFNSI	I	11		D	16					
G	2	BLDLST	B	7		J	11		E	16					
H	2		C	7		K	11		F	16					
I	2		D	7		L	11		G	16					
J	2		E	7		M	11		H	16	CONVER				
K	2		F	7		N	11		I	16					
L	2		G	7		B	12		J	16					
M	2		H	7		C	12		K	16					
N	2		I	7		D	12		L	16					
B	3		J	7		E	12		M	16					
C	3		K	7		F	12								
D	3		L	7	CLRCOR	G	12	CMPRSS							
E	3		M	7		H	12								
F	3	BRKCHK	N	7		I	12								
G	3		B	8		J	12								
H	3		C	8		K	12								
I	3		D	8		L	12								
J	3		E	8		M	12								
K	3		F	8		N	12								
L	3		G	8		B	13								
M	3		H	8		C	13								
N	3		I	8		D	13								
B	4		J	8		E	13								
C	4		K	8		F	13								
D	4		L	8	CLRSI	G	13	CMPTRT							
E	4		M	8		H	13								
F	4	CARVE	N	8		I	13								
G	4		B	9		J	13								
H	4		C	9		K	13								
I	4		D	9		L	13								
J	4		E	9		M	13								
K	4		F	9		N	13								
L	4		G	9		B	14								
M	4		H	9		C	14								
N	4		I	9		D	14								
B	5		J	9		E	14								
C	5		K	9	CLSFIL	F	14								
D	5		L	9		G	14	COMAND							
E	5	CHRTAB	M	9		H	14								
F	5		N	9		I	14								
G	5		B	10		J	14								
H	5		C	10		K	14								
I	5		D	10		L	14								

```
                                CONVERT::
000000                                  PUSH    <R3,R1,R2>              ;SAVE REGISTERS
000000                                  MOV     SI.VAL(R2),R0           ;POINT TO ASB
000006  016200  177776                  MOVB    \.BITS(R0),R3           ;GET THE TYPE OF SOURCE
000012  116003  000003                  MOV     R3,R2                   ;Copy the bits
000016  010302                          BIC     #^C<A.NUM>,R3           ;CLEAR GARBAGE BITS
000020  042703  177771                  PUSH    R1                      ;STORE NEW TYPE
000024                                  CMP     R1,R3                   ;Which is "larger"?
000026  020103                          BGT     1$                      ;Result
000030  003007                          BLT     7$                      ;Source
000032  002405                          BITB    #A.TMP+A.CON,R2         ;Is the source temporary or code string?
000034  132702  000030                  BNE     1$                      ;No, Have to do the conversion
000040  001003                          TST     (SP)+                   ;Yes, Clean up the stack
000042  005726                          BR      6$                      ;And do a quick return
000044  000461                  7$:     MOV     R3,(SP)                 ;No, source
000046  010316                  1$:     ASL     (SP)                    ; TIMES 2
000050  006316                          ASL     (SP)                    ; TIMES 4
000052  006316                          BEQ     2$                      ;Result, source are character
000054  001404                          TST     R1                      ;Is either
000056  005701                          BEQ     4$                      ;    one
000060  001435                          TST     R3                      ;        of them
000062  005703                          BEQ     4$                      ;            character?
000064  001433                  2$:     ADD     (SP),R3                 ;Lookup conversion
000066  061603                          MOV     MONCT(R3),MONCVT        ;    routine
000070  016337  0000006 003132'         ADD     (SP)+,R1                ;Lookup conversion
000076  062601                          MOV     RESCT(R1),RESCVT        ;    routine
000100  016137  0000006 003152'         MOV     2.(SP),R1               ;Result type is ...
000106  016601  000002                  MOV     (R0),R2                 ;POINT TO THE VALUE
000112  011002                          CALL    GETSM3                  ;TIME TO COPY
000114                                  MOV     A.ALEN(R0),R3           ;# OF ELEMENTS IN RESULT
000120  016003  000004                  BEQ     8$                      ;No elements to convert
000124  001416
000126                          3$:     CALL    @MONCVT                 ;Fetch source
000132  102410                          BVS     4$                      ;Conversion Error
000134  013737  003134' 003154'         MOV     MONVAL,RESVAL           ;Copy to
000142                                  STF     AC1,AC0                 ;    Result
000150                                  CALL    @RESCVT                 ;Store result
000154  104417                  4$:     DOMERR                          ;Yes, conversion is fatal
000156                                  SOB     R3,3$                   ;Loop for each element
                                ;
                                ; FINISH UP CONVERSION
                                ;
000162  010503                  8$:     MOV     SI,R3                   ;COPY PTR TO SI
000164  011602                          MOV     @SP,R2                  ;AND PTR TO OLD FRAME
000166  012700  000002                  MOV     #<-SI.BIT>/2,R0         ;LOOP LENGTH
000172  014301                  5$:     MOV     -(R3),R1                ;EXCHANGE
000174  014213                          MOV     -(R2),(R3)              ;    THE
000176  010112                          MOV     R1,(R2)                 ;        SI FRAMES
000200                                  SOB     R0,5$                   ;LOOP
000204                                  CALL    KILL                    ;DIE! DIE! YOU ALL DIE HORRIBLE DEATHS!
000210                          6$:     POP     <R2,R1,R3>              ;GET REGISTERS
000216  016200  177776                  MOV     SI.VAL(R2),R0           ;POINT TO RESULT BLOCK
000222                                  RETURN
```

```
CONVER  MACRO M1113  08-FEB-80 11:22  PAGE 47
CONVERT          CHANGE TYPE OF DATA IN SI FRAME

      1          000001                          .END
```

CONVER MACRO M1113 08-FEB-80 11:22 PAGE 47-1
SYMBOL TABLE

```
AC0     =%000000          CHRLBR= 000073    C.BLOK= 000064    D.DECD= 000002    EIS    = 000000
AC1     =%000001          CHRLPR= 000072    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC2     =%000002          CHRMNS= 000137    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC3     =%000003          CHRMUL= 000075    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC4     =%000004          CHRNEG= 000100    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC5     =%000005          CHRPDL= 000271    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R     025
ALLINT  003274R     025   CHRPLU= 000055    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R     025
ARYRET  003276R     025   CHRPWI  000120    C.DIG = 000040    D.EXPD= 000002    ERR0    004774R     025
ASCQUO= 000042            CHRQST= 000121    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R     025
A.ALEN= 000004            CHRQUO= 000113    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R     025
A.ARG = 000040            CHRRBR= 000047    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R     025
A.ASB = 000200            CHRRPR= 000042    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.BITS= 000003            CHRSEM= 000074    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.CON = 000020            CHRSMO= 000112    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIMS= 000002            CHRSPA= 000040    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.DIM1= 000006            CHRUPA= 000131    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.NUM = 000006            CHR.BS= 000010    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.PTR = 000000            CHR.CC= 000003    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
A.TMP = 000010            CHR.CR= 000015    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADDIR= 104463            CHR.CZ= 000032    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BADERR= 104437            CHR.GT= 000046    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BLZBUB  005116R     025   CHR.LF= 000012    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
BRSELF  005046R     025   CHR.LT= 000043    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CCNT    005072R     025   CHR.SI= 000017    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHANEX  003060R     025   CHR.SL= 000057    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN00  000256R     025   CHR.SO= 000016    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN01  000330R     025   CHR.TB= 000011    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN02  000512R     025   CHR.UL= 000106    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN03  000674R     025   CHR..A= 000141    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN04  001056R     025   CHR..E= 000145    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN05  001240R     025   CHR..S= 000163    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN06  001422R     025   CHR..T= 000164    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN07  001604R     025   CHR..U= 000165    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN08  001766R     025   CHR..Z= 000172    DEVIL   005114R     025 D.QUAD= 000000    FD.F11= 040000
CHAN09  002150R     025   CHR..0= 000060    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN10  002332R     025   CHR..1= 000061    DIGITS  000006R     025 D.RHO = 000000    FD.ISP= 002000
CHAN11  002514R     025   CH.AND= 000001    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN12  002676R     025   CMDBUF  003250R     025 DIMBLK  003112R     025 D.SCON= 000006    FD.OSP= 004000
CHAN13  000022R     023   CMDMAX= 000010    DIMPTR  003114R     025 D.SDIF= 000006    FD.PLC= 000004
CHAN14  000204R     023   CNTSIZ  005130R     025 DIMTYP  003116R     025 D.SUB = 000000    FD.PRN= 000004
CHNOPN= 104471            COMBIN  005076R     025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051            COMCNT  005126R     025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007            COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133            COMPTR  005122R     025 DYDCNT  003224R     025 D.UNON= 000006    FD.REC= 000001
CHRBOL= 000200            COMRTN  003156R     025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114            COMSKP  005124R     025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054            CONVER  000000RG    020 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076            CORPTR  005120R     025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107            CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053            CS.BOO= 000010    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176            CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110            CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056            CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105            CTEPS   000152R     025 D.COMP= 000001    EDICHR  005063R     025 FF.RWF= 000006
CHRERR= 104431            CTMAX   000212R     025 D.CONT= 000006    EDICNT  005064R     025 FF.SPC= 000004
CHRFIL= 000177            CTRMAX  000172R     025 D.CTLM= 000000    EDIFLG  005062R     025 FHEADR  005012R     025
CHRFRO= 000135            C.ALOC= 000162    D.DEAL= 000003    EDIT   = 000003          FHRET   005040R     025
```

CONVER MACRO M1113 08-FEB-80 11:22 PAGE 47-2
SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ICHAN   005050R     025 MONCVT  003132R     025 M.REVS= 000003
FIS    = 000000          F.FNDF= 000010          ILFNAM= 104464          MONLEN  003130R     025 M.ROLL= 000000
FISCON= 046000          F.FTYP= 000116          ILLCHN= 104540          MONPTR  003122R     025 M.SATA= 000001
FIXPTR  003074R     025 F.FUN = 000010          ILLSW = 104565          MONSFT  003126R     025 M.SCAN= 000011
FLAGS   004764R     025 F.FVER= 000120          INDERR= 104410          MONTYP  003124R     025 M.SHAP= 000000
FLCBUF  003250R     025 F.HEAD= 000006          INNTMP  003250R     025 MONVAL  003134R     025 M.SIGN= 000000
FLINE   005014R     025 F.HIBK= 000004          INNTM2  005016R     025 M.ABS = 000000          M.SUBR= 000001
FLOEXP  003270R     025 F.LBL = 000020          INPTR   000064R     025 M.BOL = 000200          M.SYM = 000203
FLONEG  003272R     025 F.LINE= 000012          INSNAM= 104462          M.BXAV= 000007          M.SYN = 000201
FLSWTB  000222R     025 F.LOC = 000004          IOERR = 104477          M.BXCR= 000007          M.TRAN= 000012
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776          M.BXCT= 000007          M.$ABS= 000000
FNINSI  005044R     025 F.LUN = 000042          I.DIM = 177772          M.BXEX= 000007          M.$CEI= 000000
FORIGN  000032R     025 F.MBCT= 000054          I.LEN = 000012          M.BXFX= 000007          M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062          M.BXIO= 000007          M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770          M.BXLC= 000003          M.$NEG= 000000
FO.RD = 000001          F.MON = 000002          I.PTR = 177774          M.BXNC= 000007          M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766          M.BXNL= 000007          M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024          KILL   = ****** GX      M.BXPP= 000007          M.1REV= 000003
FPMODE  000042R     025 F.NREC= 000030          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FPP    = 000000          F.OPN = 000010          LABLPT  005042R     025 M.BXRL= 000007          NAMBUF  003240R     025
FPPLEN= 000004          F.OVBS= 000030          LARGBL  003162R     025 M.BXTS= 000007          NB.DEV= 000200
FREBLK  007400R     025 F.PREV= 000000          LARGCV  003176R     025 M.BXWA= 000007          NB.DIR= 000100
FUZZ    000142R     025 F.PRNT= 100000          LARGFL  003160R     025 M.CEI = 177777          NB.NAM= 000004
FXPTR1  003076R     025 F.PROT= 010000          LARGLE  003174R     025 M.CHR = 000202          NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016          LARGLN  003166R     025 M.CMT = 000206          NB.SD2= 001000
F$EVI  = 000200          F.RATT= 000001          LARGPT  003164R     025 M.DOUT= 000006          NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034          LARGSF  003172R     025 M.DQAD= 000006          NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017          LARGTY  003170R     025 M.ENCD= 000004          NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000          LARGVA  003200R     025 M.EXE = 000000          NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040          LBINC   005100R     025 M.EXP = 000001          NB.VER= 000001
F.BBFS= 000062          F.RES = 000040          LBUFF   003316R     025 M.FACT= 000001          ND.DEV= 000200
F.BDB  = 000070          F.RET = 177772          LBUFLN= 000621          M.FIL = 000177          ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002          LCOUNT  005106R     025 M.FLR = 177777          ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000          LENERR= 104412          M.FMT = 000004          ND.PPN= 000100
F.BKDN= 000026          F.R2   = 000014          LENTAB  000054R     025 M.GDUP= 000003          ND.$EX= 000000
F.BKDS= 000020          F.R3   = 000016          LEXPTR  003100R     025 M.GDWN= 000003          ND.$ES= 000020
F.BKEF= 000050          F.R4   = 000020          LFTOP   005066R     025 M.GETI= 000006          ND.$NM= 000000
F.BKP1= 000051          F.SEQN= 000100          LINC    005102R     025 M.IBM = 000003          ND.$NS= 000040
F.BKST= 000024          F.SPDV= 000072          LINE    005016R     025 M.IDN = 000000          ND.$PN= 001400
F.BKVB= 000064          F.SPUN= 000074          LININC  005026R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BR   = 000020          F.STBK= 000036          LISIZE= 000020          M.IOTA= 000000          NINPT1  003302R     025
F.CHR  = 000075          F.STK  = 000022          LITOP   005072R     025 M.LIT = 000205          NINPT2  003300R     025
F.CNTG= 000034          F.STOP= 000004          LNLNG  = 104541          M.LN   = 000001          NINSTK  003304R     025
F.CODE= 000010          F.STP  = 000001          LOCLCT  005036R     025 M.MMD = 000010          NINTMP  003310R     025
F.DFNB= 000046          F.SUSP= 000002          LOWCOR  000002R     025 M.NEG = 000000          NINTYP  003306R     025
F.DSPT= 000044          F.TRC  = 000002          LSTERP  004766R     025 M.NOT = 000000          NOTAVL= 104472
F.DVNM= 000134          F.UNIT= 000136          LWIDTH  000070R     025 M.NUB = 000006          NOTOPN= 104473
F.EFBK= 000010          F.URBD= 000020          L.BITS= 000003          M.NUM = 000204          NPRT1   004762R     025
F.EFN  = 000050          F.VAL  = 177776          L.CNT  = 000002          M.OUTR= 000012          NULL    007364R     025
F.END  = 002000          F.VBN  = 000064          L.PRNT= 000001          M.PITM= 000000          NUMCHN= 000020
F.ECBB= 000032          F.VBSZ= 000060          L.PTR  = 000000          M.PRDS= 000006          NUMDFL  003230R     025
F.ERR  = 000052          GETSM3= ****** GX       L.PTR1= 000004          M.QQAD= 000000          NUMEFL  003233R     025
F.FACC= 000043          GNOME   003102R     025 L.SUBS= 000002          M.QUAD= 000000          NUMESN  003236R     025
F.FFBY= 000014          HASHC   005136R     025 MACHIN= 000106          M.RAND= 000012          NUMFLG  003230R     025
F.FLAG= 000002          HASHTB  005140R     025 MAXCHN= 000014          M.RAVL= 000000          NUMLEN  000062R     025
F.FLB  = 000100          HSHMSK= 177601          MONBLK  003120R     025 M.RCIP= 000000          NUMNEG  003232R     025
F.FNAM= 000110          HSHSIZ= 000200          MONCT  = ****** GX      M.RDCT= 000011          NUMQFL  003234R     025
```

CONVER MACRO M1113 08-FEB-80 11:22 PAGE 47-3
SYMBOL TABLE

```
NUMRIG  003231R     025 RANERR= 104411          R.NEXT= 000006          SY.BS  = 000010          T.LIT  = 000020
NUMWID  003226R     025 RARGBL  003204R    025 R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
NUMZFL  003235R     025 RARGCV  003220R    025 R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
NYIERR= 104432          RARGFL  003202R    025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
N.BLSZ= 000036          RARGLE  003216R    025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLN  003210R    025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGPT  003206R    025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DEVN= 000034          RARGSF  003214R    025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGTY  003212R    025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.DVNM= 000032          RARGVA  003222R    025 SIDAMN= 104433           S.AB   = 177776          T.VAL  = 100000
N.EXT = 000014          RBASE   005074R    025 SIFRAM  003314R    025 S.ASB  = 100000          T.VAR  = 001000
N.FID = 000000          RCOUNR  005112R    025 SISTAK  006350R    025 S.BFHD= 000020          UPPER  = 177400
N.FNAM= 000006          RCOUNT  005110R    025 SITOP   006346R    025 S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          REDCNT  005070R    025 SITP    007144R    025 S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDIGN  005066R    025 SIWARN= 000200           S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDOFF  005072R    025 SIZTAB  000044R    025 S.FDB  = 000140          VARBLS  000000R     025
N.NEXT= 000022          REENTR  003060R    025 SI.BIT= 177774           S.FLB  = 040000          VERNUM  000000R     025
N.PPN = 000024          RESBIT  003142R    025 SI.LEN= 177772           S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESCT = ****** GX       SI.RET= 177772           S.FNB  = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESCVT  003152R    025 SI.VAL= 177776           S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESERR= 104440          SKPSIZ  005134R    025 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESLEN  003150R    025 SPHOLD  000004R    024 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESPTR  003140R    025 SPINIT  000002R    024 S.LEN  = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESSFT  003146R    025 STKINC= 000774           S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESTYP  003144R    025 STOP   = 104420          S.NFEN= 000020          $FIS   = ****** G
OPTMP2  005072R     025 RESULT  003136R    025 SUBCNT  003104R    025 S.TYPE= 140000          $FPP   = ****** G
OPTMP3  005074R     025 RESVAL  003154R    025 SUBDCT  003106R    025 S.VAL  = 000000          $FPPLN= ****** G
OPTMP4  005076R     025 RINC    005104R    025 SUBPTR  003110R    025 TKDIMS  005072R    025 $MACHN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R    025 SYMB1   005344R    025 TKFRAM  005066R    025 $RSTS  = ****** G
OPTMP6  005102R     025 ROTINC  005070R    025 SYMF    005340R    025 TKLEFT  005070R    025 $RSX   = ****** G
OPTMP7  005104R     025 ROUND   000010R    025 SYMPT1  005346R    025 TMODE   000403R    023 $RT11  = ****** G
OPTMP8  005106R     025 ROVER   003312R    025 SYMTAB  005352R    025 TMPTAG= 000166          $STF   = ****** GX
OPTMP9  005110R     025 RSTS   = 000000         SYMTBF= 104404           TTYBUF  004140R    025 .BOOLE= 000002
OPTM10  005112R     025 RSX    = 000115         SYMTOP  005350R    025 TTYLEN= 000622          .CHARA= 000000
OPTM14  005026R     025 RTOP    005070R    025 SYMTP   006272R    025 TURKEY= 104436          .DEF.  = 000002
ORIGIN  000030R     025 RT11   = 000000         SYMWRN= 000054           TYPE   = 000166          .FLOAT= 000006
OUTPTR  000066R     025 R.BITS= 000003          SYNERR= 104407           T.BOL  = 020000          .FSRCB= ****** G
OVROPR= 000000          R.FIX  = 000001         SYSERR= 104402           T.CON  = 004000          .INTEG= 000004
PATCH   007344R     025 R.LCT  = 000002         SYSTAT  000000R    024 T.DIM  = 000100          .MOLUN  000404R     023
PRECSN  000052R     025 R.LN   = 000002         SY.BAD= 000001           T.FUN  = 010000          ...GBL= 000000
RANDAC  003064R     025 R.LNUM= 000010          SY.BRK= 100000           T.LIST= 000400           ...TPC= 000020
RANDY   003062R     025

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

```
CONVER  MACRO M1113  08-FEB-80 11:22  PAGE 47-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000224      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12953 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:47
CONVER,DM1:'305,301]CONVER/-SP=APLRX1,APLDEF,CONVER,END
```

FILEID**COPYC

```
  CCCCCCCC      000000      PPPPPPPP      YY      YY      CCCCCCCC
  CCCCCCCC      000000      PPPPPPPP      YY      YY      CCCCCCCC
CC            OO      OO    PP      PP    YY      YY    CC
CC            OO      OO    PP      PP    YY      YY    CC
CC            OO      OO    PP      PP      YY  YY      CC
CC            OO      OO    PP      PP      YY  YY      CC
CC            OO      OO    PPPPPPPP          YY        CC
CC            OO      OO    PPPPPPPP          YY        CC
CC            OO      OO    PP                YY        CC
CC            OO      OO    PP                YY        CC
CC            OO      OO    PP                YY        CC                  ....
CC            OO      OO    PP                YY        CC                  ....
  CCCCCCCC      000000      PP                YY          CCCCCCCC          ....
  CCCCCCCC      000000      PP                YY          CCCCCCCC          ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:24
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CPINAM           )COPY - INPUT NAME STRING
```

COPYC MACRO M1113 08-FEB-80 11:24
TABLE OF CONTENTS

45-	55	CPONAM	)COPY - OUTPUT NAME STRING
46-	99	OWORD	OUTPUT A WORD TO THE CURRENT OUTPUT CHANNEL
46-	123	GETWRD	INPUT A WORD FROM THE CURRENT INPUT CHANNEL
47-	148	COPYC	

COPYC MACRO M1113 08-FEB-80 11:24 PAGE 1

```
                                RT11    =  0            ;NOT RT-11
                                RSTS    =  0            ;NOT RSTS/E
                                RSX     = 'M            ;RSX-11/M

                                EIS     =  0            ;NO EIS
                                FIS     =  0            ;NO FIS
                                FPP     =  0            ;NO FPP

                                FPPLEN  =  4.           ;SINGLE PRECISION

                                DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
COPYC   MACRO M1113  08-FEB-80 11:24  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                        .TITLE  COPYC
                        .IDENT  \V02.03\
                        .SBTTL  C.INAM          )COPY - INPUT NAME STRING
                        ;
                        ; FUNCTIONAL DESCRIPTION:
                        ;       This routine inputs a name in the "standard" copy
                        ;       format to the currently open channel. The format is
                        ;
                        ;       ******************
                        ;       *  Name Length   *     (Words)
                        ;       ******************
                        ;       * Chr 1 * Length*
                        ;       ******************
                        ;       *  ...  * Chr 2 *
                        ;              .
                        ;              .
                        ;              .
                        ;       ******************
                        ;
                        ; IMPLICIT INPUTS:
                        ;       This routine inputs the name specified to the current
                        ;       output channel.
                        ;
                        ; OUTPUTS:
                        ;       R0      - Pointer to Name entry in Symbol Table
                        ;
                        ; ROUTINES CALLED:
                        ;       GETBLK  - Input a # of bytes from the input file
                        ;--
                        .PSECT  CODE

                        CPINAM::
                                PUSH    <R1,R2,R3,R4>   ;Save most registers
                                CALL    GETWRD          ;Get the Length
                                MOV     R0,R4           ;Copy size (in bytes)
                                BEQ     1$              ;Nothing there, Finished
                                MOV     #NAMBUF,R1      ;Set up Pointer to the Name Buffer
                                CALL    GETBLK          ;Get a block of characters
                                MOV     #NAMBUF,R4      ;Pointer to Name Buffer (again)
                                MOVB    (R4)+,R3        ;Name length (# of chars)
                                BLE     2$              ;Really shouldn't happen
                                MOVB    (R4)+,R2        ;Get the next character
                                DEC     R3              ;   and decrement count remaining
                                CALL    SYMBOL          ;Find it (or put it) in the symbol table
                        1$:     POP     <R4,R3,R2,R1>   ;Restore the registers
                                TST     R0              ;See if returned a Pointer
                                RETURN

                        2$:     SYSERR                  ;Really shouldn't happen
```

```
                                .SBTTL  CPONAM          )COPY - OUTPUT NAME STRING

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine outputs a name in the ''standard'' copy
                                ;       format to the currently open channel. The format is
                                ;
                                ;       ****************
                                ;       *  Name Length  *     (Words)
                                ;       ****************
                                ;       * Chr 1 * Length*
                                ;       ****************
                                ;       *  ...  * Chr 2 *
                                ;                 .
                                ;                 .
                                ;                 .
                                ;       ****************
                                ;
                                ; INPUTS:
                                ;       R1      - Pointer to Name to output
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       This routine outputs the name specified to the current
                                ;       output channel.
                                ;
                                ; ROUTINES CALLED:
                                ;       OUTBLK  - Output the name block
                                ;       OWORD   - Output the size, in bytes
                                ;--
                                        .PSECT  CODE

000066                          CPONAM::
000066                                  PUSH    <R4,R1>         ;Save work registers
000072  005721                          TST     (R1)+           ;Point to the length
000074  111104                          MOVB    (R1),R4         ;Get size, in bytes, of the name
000076  062704  000002                  ADD     #2.,R4          ;Add in 2 for overhead
000102  042704  000001                  BIC     #<1>,R4         ;Clear the low bit (Copy words)
000106  010400                          MOV     R4,R0           ;Copy the size
000110                                  CALL    OWORD           ;Output, no translation
000114                                  CALL    OUTBLK          ;Output a block
000120                                  POP     <R1,R4>         ;Restore the registers
000124                                  RETURN                  ;Finished
```

```
                        .SBTTL  OWORD           OUTPUT A WORD TO THE CURRENT OUTPUT CHANNEL

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       This routine outputs a word to the currently open
                        ;       output channel.
                        ;
                        ; INPUTS:
                        ;       R0      - The word to be output
                        ;
                        ; ROUTINES CALLED:
                        ;       OCHAR   - Outputs a byte
                        ;--
000126                          .PSECT  CODE

000126                  OWORD::
000126                          CALL    OCHAR           ;Output the lower half
000132  000300                  SWAB    R0              ;Switch bytes
000134                          CALL    OCHAR           ;Output the Upper half
000140  000300                  SWAB    R0              ;Back as before
000142                          RETURN                  ;Finished


                        .SBTTL  GETWRD          INPUT A WORD FROM THE CURRENT INPUT CHANNEL

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       This routine inputs a word from the currently open
                        ;       input channel.
                        ;
                        ; INPUTS:
                        ;       R0      - The word to be input
                        ;
                        ; ROUTINES CALLED:
                        ;       GETBYT  - inputs a byte
                        ;--
000144                          .PSECT  CODE

000144                  GETWRD::
000144                          PUSH    R2              ;Save R2
000146  013702  005050'         MOV     ICHAN,R2        ;Get the input channel
000152                          CALL    GETBYT          ;input the lower half
000156                          PUSH    R0              ;Save the character
000160                          CALL    GETBYT          ;input the Upper half
000164  110066  000001          MOVB    R0,1.(SP)       ;Put in top byte
000170                          POP     <R0,R2>         ;Get the word
000174                          RETURN                  ;Finished
```

```
                                .SBTTL  COPYC

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is used to perform the copy command. It does
                                ;       a whole mess of stuff that I'll describe in the code.
                                ;--
000176                          .PSECT  CODE
                                .ENABL  LSB

000176                          PCOPYC::
000176  012737  000001  005070' MOV     #1.,OPTMP1              ;Indicate PCOPY
000204  000402                 BR      1$                      ;   and go

000206                          COPYC::
000206  005037  005070'         CLR     OPTMP1                  ;Indicate Copy
                                ;+
                                ; Take care of scanning the file name, putting it in a
                                ; file name descriptor
                                ;-
000212  162706  000006         1$:     SUB     #6.,SP                  ;Reserve space on SP stack
000216                                  PUSH    <R4,R3>                 ;Save the current pointers
000222                                  CALL    GETNB                   ;Get 1st non-blank character
000226                         2$:      GETC    3$                      ;Get another character
000226  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000230  002404                          BLT 3$                  ;NO, BRANCH TO 3$
000232  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000234  122702  000040                  CMPB    #CHRSPA,R2              ;Is it a space?
000240  001372                          BNE     2$                      ;No, continue
000242  010466  000004          3$:     MOV     R4,4.(SP)               ;Save current string info
000246  010366  000006                  MOV     R3,6.(SP)               ;   (all three
000252  010266  000010                  MOV     R2,8.(SP)               ;       registers)
000256  005103                          COM     R3                      ;Make it negative
000260  062603                          ADD     (SP)+,R3                ;   to get string length
000262                                  POP     R4                      ;Restore address, too
000264  012702  000000G                 MOV     #DFLTSV,R2              ;Point to default name block
000270                                  CALL    GETNAM                  ;Decode the file name
                                ;+
                                ; We must save the current file for later
                                ;-
000274  012702  000022'                 MOV     #CHAN13,R2              ;Utility channel
000300  052762  000400  177756          BIS     #C.NOBF,C.SBIT(R2)      ;Normal buffering
000306  012701  177775                  MOV     #-3.,R1                 ;Parameters for OPNSET
000312                                  CALL    OPNSET                  ;Open a temporary file
000316  012700  000124'                 MOV     #CHAN13+C.NAME,R0       ;Pointer to FID field
000322  012701  000344'                 MOV     #UTNAME,R1              ;Place to save it
000326  012703  000017                  MOV     #N.BLSZ/2,R3            ;Length to copy
000332  012021                  4$:     MOV     (R0)+,(R1)+             ;Word by word
000334                                  SOB     R3,4$                   ;Copy it all
000340                                  CALL    SAVOVR                  ;Save the WS
000344  052737  000004  000000'         BIS     #SY.SVC,SYSTAT          ;Copy in progress
```

```
                                        ;+
                                        ; Set up the temporary file for outputting the names
                                        ;-
000352  012701  177775                          MOV     #-3.,R1                 ;Temporary file
000356  012702  000022'                        MOV     #CHAN13,R2              ;   on Utility Channel #1
000362                                          CALL    OPNSET                  ;Ask for a temporary
000366  010200                                  MOV     R2,R0                   ;Point to the FDB
000370                                          CALL    .MRKDL                  ;Mark file for deletion
000374  010237  005052'                        MOV     R2,OCHAN                ;Channel for output
                                        ;+
                                        ; Now we scan the input line, and output the values.
                                        ;-
000400  005000                          5$:     CLR     R0                      ;Assume no elements
000402                                          POP     <R4,R3,R2>              ;Restore string info registers
000410                                          CALL    GETNB2                  ;Get the next non-blank character
000414  002411                                  BLT     5$                      ;Nothing Specified, copy all
000416  122702  000015                          CMPB    #CHR.CR,R2              ;Hello? End of Line?
000422  001406                                  BEQ     5$                      ;Yes, same thing
000424                                          CALL    BLDLST                  ;Build a list of the stuff
000430  016502  177776                          MOV     SI.VAL(SI),R2           ;Get a pointer to the list
000434  116200  000002                          MOVB    L.CNT(R2),R0            ;Get the # of names
000440                                  5$:     PUSH    R0                      ;Save # of names
000442                                          CALL    OWORD                   ;Output count
000446                                          POP     R3                      ;Anything there?
000450  001406                                  BEQ     7$                      ;No, fini'
000452  011202                                  MOV     (R2),R2                 ;Point to element
000454  012201                          6$:     MOV     (R2)+,R1                ;Pointer to name
000456                                          CALL    CPONAM                  ;Output the name
000462                                          SOB     R3,6$                   ;Loop for every element
000466  012702  000022'                  7$:     MOV     #CHAN13,R2              ;Point to Utility Channel #1
000472                                          CALL    WRITEB                  ;Write it out
000476  005062  177760                          CLR     C.BUFF(R2)              ;No more buffer, am loading new WS
```

```
                                ;+
                                ; Load in the WS we will copy from, read the names, output the
                                ; corresponding data, read in the name block of the temporary
                                ;-
000502  012702  000204'         MOV     #CHAN14,R2              ;Point to Utility Channel #2
000506  005001                  CLR     R1                      ;File Should exist
000510  052762  000400  177756  BIS     #C.NOBF,C.SBIT(R2)      ;Indicate no buffer
000516                          CALL    OPNSET                  ;Ask for it
000522                          CALL    LODOVR                  ;Load the file
000526  052737  000004  000000' BIS     #SY.SVC,SYSTAT          ;Copy in progress
000534  012702  000204'         MOV     #CHAN14,R2              ;Temp file for data
000540  012701  177775          MOV     #-3.,R1                 ;It is a temporary file
000544                          CALL    OPNSET                  ;Get the file
000550  010200                  MOV     R2,R0                   ;Point to the FDB
000552                          CALL    .MRKDL                  ;Mark file for deletion
000556  010237  005052'         MOV     R2,OCHAN                ;This is the output file
000562  012702  000022'         MOV     #CHAN13,R2              ;Input channel
000566  005062  000064          CLR     F.BKVB(R2)              ;Point to
000572  012762  000001  000066  MOV     #1.,F.BKVB+2(R2)        ;   beginning of file
000600  010237  005050'         MOV     R2,ICHAN                ;Now input file
000604                          CALL    READB                   ;Now read it in
000610  005062  000064          CLR     F.BKVB(R2)              ;Point to
000614  012762  000001  000066  MOV     #1.,F.BKVB+2(R2)        ;   beginning of file
                                ;
                                ; Copy the date the WS was saved
                                ;
000622  012701  000136'         MOV     #WSDATE,R1              ;Point to WS date
000626  012704  000003          MOV     #3.,R4                  ;Date Info
000632                          CALL    OUTBLK                  ;Output it
000636  012701  000132'         MOV     #WSTIME,R1              ;Point to WS time
000642  012704  000003          MOV     #3.,R4                  ;Time info
000646                          CALL    OUTBLK                  ;Output it

000652                          CALL    CPOARG                  ;Copy out the stuff
000656  013702  005050'         MOV     ICHAN,R2                ;Point to input channel
000662                          CALL    RESCHN                  ;Reset it
000666  013702  005052'         MOV     OCHAN,R2                ;Point to output stuff
000672                          CALL    WRITEB                  ;Write out last buffer
000676  005062  177760          CLR     C.BUFF(R2)              ;Clear buffer pointer
                                ;
                                ; Re-load the old WS
                                ;
000702  012700  000344'         MOV     #UTNAME,R0              ;Pointer to Save area
000706  012703  000124'         MOV     #CHAN13+C.NAME,R3       ;Pointer to channel block
000712  012701  000017          MOV     #N.BLSZ/2,R1            ;# of words to copy
000716  012023          9$:     MOV     (R0)+,(R3)+             ;Word copy
000720                          SOB     R1,9$                   ;   real fast
000724  012702  000022'         MOV     #CHAN13,R2              ;Point to the channel
000730  052762  000400  177756  BIS     #C.NOBF,C.SBIT(R2)      ;Indicate no buffer
000736                          CALL    OPNSET                  ;Open the file
000742  010200                  MOV     R2,R0                   ;Point to the FDB
000744                          CALL    .MRKDL                  ;Mark file for deletion
000750                          CALL    LODOVR                  ;Load the file
000754  062706  000006          ADD     #6.,SP                  ;Pop arguments we don't need
```

```
                                ;
                                ; Copy the data from the temporary file into the system
                                ;
000760  012702  000204'            MOV     #CHAN14,R2              ;Input channel
000764  010237  005050'            MOV     R2,ICHAN                ;This is the Input file
000770  005062  000064             CLR     F.BKVB(R2)              ;Point to
000774  012762  000001  000066     MOV     #1.,F.BKVB+2(R2)        ;   beginning of file
001002                             CALL    READB                   ;Now read it in
001006  005062  000064             CLR     F.BKVB(R2)              ;Point to
001012  012762  000001  000066     MOV     #1.,F.BKVB+2(R2)        ;   beginning of file
001020  012701  000136'            MOV     #WSDATE,R1              ;Point to the date
001024  012704  000003             MOV     #3.,R4                  ;Read in 3. bytes
001030                             CALL    GETBLK                  ;Read in a block
001034  012701  000132'            MOV     #WSTIME,R1              ;Point to the time
001040  012704  000003             MOV     #3.,R4                  ;Read in 3. bytes
001044                             CALL    GETBLK                  ;Get the time

001050                             CALL    CPIARG
001054  013702  005050'            MOV     ICHAN,R2                ;Point to the input channel
001060                             CALL    RESCHN                  ;Close the channel
001064                             MPRINT  SAVMSG                  ;"SAVED "
001100                             CALL    DATIME                  ;Print the date/time
001104                             CALLR   NEWLIN                  ;   and a <CR><LF>

                                   .DSABL  LSB
000000                             .PSECT  TABLES

000000      163     141     166  SAVMSG: .ASCIZ  /saved /
000003      145     144     040
000006      000

                                   .EVEN
```

```
COPYC   MACRO M1113  08-FEB-80 11:24  PAGE 52
COPYC

      1          000001                          .END
```

COPYC MACRO M1113 08-FEB-80 11:24 PAGE 52-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    CS.ONE= 000100      D.BXRL= 000007    D.1COM= 000001
AC1    =%000001          CHRLBR= 000073    CTEPS   000152R 025 D.CIRC= 000001    D.1EXP= 000002
AC2    =%000002          CHRLPR= 000072    CTMAX   000212R 025 D.COMB= 000001    D.1ROT= 000003
AC3    =%000003          CHRMNS= 000137    CTRMAX  000172R 025 D.COMP= 000001    EDICHR  005063R 025
AC4    =%000004          CHRMUL= 000075    C.ALOC= 000162      D.CONT= 000006    EDICNT  005064R 025
AC5    =%000005          CHRNEG= 000100    C.BLOK= 000064      D.CTLM= 000000    EDIFLG  005062R 025
ALLINT  003274R     025  CHRPDL= 000271    C.BLSZ= 000140      D.DEAL= 000003    EDIT   = 000003
ARYRET  003276R     025  CHRPLU= 000055    C.BUFF= 177760      D.DECD= 000002    EIS    = 000000
ASCQUO= 000042           CHRPWR= 000120    C.BUFL= 177762      D.DIV = 000000    ELESIZ  005132R 025
A.ALEN= 000004           CHRQST= 000121    C.CCNT= 177766      D.DMD = 000010    ENDCOR  000004R 025
A.ARG = 000040           CHRQUO= 000113    C.CPTR= 177764      D.DQAD= 177777    EOF    = 104475
A.ASB = 000200           CHRRBR= 000047    C.CZ   = 001000     D.DROP= 000002    ERRMAX= 000166
A.BITS= 000003           CHRRPR= 000042    C.DIG = 000040      D.ENCD= 000002    ERRNUM  004772R 025
A.CON = 000020           CHRSEM= 000074    C.FASW= 020000      D.EQ   = 000000   ERRPTR  004770R 025
A.DIMS= 000002           CHRSMO= 000112    C.FLEN= 000010      D.EXPD= 000002    ERR0    004774R 025
A.DIM1= 000006           CHRSPA= 000040    C.FLSH= 004000      D.FGET= 000005    EXEBUF  005054R 025
A.NUM = 000006           CHRUPA= 000131    C.FRC = 000001      D.FMT = 000004    E$CT    005056R 025
A.PTR = 000000           CHR.BS= 000010    C.IDEN= 000010      D.FPNT= 000005    E$PT    005060R 025
A.TMP = 000010           CHR.CC= 000003    C.NAME= 000102      D.FPUT= 000005    FA.APD= 000100
BADDIR= 104463           CHR.CR= 000015    C.NOBF= 000400      D.GE   = 000000   FA.CRE= 000010
BADERR= 104437           CHR.CZ= 000032    C.NUM = 000020      D.GT   = 000000   FA.DLK= 001000
BLDLST= ****** GX        CHR.GT= 000046    C.OP   = 000004     D.IBM = 000002    FA.ENB= 100000
BLZBUB  005116R     025  CHR.LF= 000012    C.OPN = 010000      D.IND = 000002    FA.EXC= 002000
BRSELF  005046R     025  CHR.LT= 000043    C.RDLN= 177776      D.INNR= 000013    FA.EXT= 000004
CCNT    005072R     025  CHR.SI= 000017    C.REWT= 002000      D.INTR= 000006    FA.NSP= 000100
CHANEX  003060R     025  CHR.SL= 000057    C.RND = 000010      D.LE   = 000000   FA.NSR= 000200
CHAN00  000256R     025  CHR.SO= 000016    C.SBIT= 177756      D.LOG = 000001    FA.POS= 010000
CHAN01  000330R     025  CHR.TB= 000011    C.SPFX= 000100      D.LT   = 000000   FA.RD  = 000001
CHAN02  000512R     025  CHR.UL= 000106    C.SPGL= 000200      D.MAX = 000000    FA.RWD= 004000
CHAN03  000674R     025  CHR..A= 000141    C.SPLX= 000001      D.MEM = 000002    FA.SEQ= 040000
CHAN04  001056R     025  CHR..E= 000145    C.SPRS= 000002      D.MIN = 000000    FA.SHR= 000040
CHAN05  001240R     025  CHR..S= 000163    C.SRC = 177774      D.MUL = 000000    FA.TMP= 000020
CHAN06  001422R     025  CHR..T= 000164    C.SRCP= 000024      D.NE   = 000000   FA.WCK= 020000
CHAN07  001604R     025  CHR..U= 000165    C.TT   = 000004     D.NN   = 000000   FA.WRT= 000002
CHAN08  001766R     025  CHR..Z= 000172    C.WIDC= 177770      D.NR   = 000000   FD.BLK= 000010
CHAN09  002150R     025  CHR..0= 000060    C.WIDR= 177772      D.OR   = 000000   FD.CCL= 000002
CHAN10  002332R     025  CHR..1= 000061    DATIME= ****** GX   D.PLUS= 000000    FD.COM= 020000
CHAN11  002514R     025  CH.AND= 000001    DEBUG = 000000      D.PSUB= 000006    FD.CR  = 000002
CHAN12  002676R     025  CMDBUF  003250R 025 DEFERR= 104405    D.PWR = 000001    FD.DIR= 000010
CHAN13  000022R     023  CMDMAX= 000010      DEVIL   005114R 025 D.QQAD= 000000    FD.FTN= 000001
CHAN14  000204R     023  CNTSIZ  005130R 025 DEVNTF= 104470      D.QUAD= 000000    FD.F11= 040000
CHNOPN= 104471           COMBIN  005076R 025 DFLTSV= ****** GX   D.RES = 000000    FD.INS= 000010
CHRAND= 000051           COMCNT  005126R 025 DIGITS  000006R 025 D.RHO = 000000    FD.ISP= 002000
CHRBEL= 000007           COMERR= 104425      DIMAX = 000020      D.ROT = 000003    FD.MNT= 100000
CHRBKR= 000133           COMPTR  005122R 025 DIMBLK  003112R 025 D.SCON= 000006    FD.OSP= 004000
CHRBOL= 000200           COMRTN  003156R 025 DIMPTR  003114R 025 D.SDIF= 000006    FD.PLC= 000004
CHRBOX= 000114           COMSKP  005124R 025 DIMTYP  003116R 025 D.SUB = 000000    FD.PRN= 000004
CHRCMA= 000054           COPYC   000206RG 020 DIRFUL= 104466     D.SUBS= 000006    FD.PSE= 010000
CHRCOL= 000076           CORPTR  005120R 025 DOMERR= 104417      D.TAKE= 000002    FD.RAH= 000001
CHRDEL= 000107           CPIARG= ****** GX   DPTERR= 104416      D.TRAN= 000004    FD.RAN= 000002
CHRDIV= 000053           CPINAM  000000RG 020 DYDCNT  003224R 025 D.UNON= 000006    FD.REC= 000001
CHRDLR= 000176           CPOARG= ****** GX   D.AND = 000000      D.$MAX= 000000    FD.RWM= 000001
CHRDLT= 000110           CPONAM  000066RG 020 D.BXCT= 000007     D.$MIN= 000000    FD.SDI= 000020
CHRDOT= 000056           CS.ASS= 000200      D.BXIO= 000007      D.$MUL= 000000    FD.SQD= 000040
CHREP = 000105           CS.BOO= 000010      D.BXNL= 000007      D.$PLU= 000000    FD.TTY= 000004
CHRERR= 104431           CS.FLO= 000020      D.BXPP= 000007      D.$RES= 000000    FD.WBH= 000002
CHRFIL= 000177           CS.IDN= 000040      D.BXPW= 000007      D.$SUB= 000000    FF.CHR= 000005
```

COPYC MACRO M1113 08-FEB-80 11:24 PAGE 52-2
SYMBOL TABLE

```
FF.NV = 000003          F.EOBB= 000032          F.VBSZ= 000060          LSTERP  004766R 025 M.NUB = 000006
FF.POE= 000002          F.ERR = 000052          GETBLK= ****** GX       LWIDTH  000070R 025 M.NUM = 000204
FF.RWD= 000001          F.FACC= 000043          GETBYT= ****** GX       L.BITS= 000003      M.OUTR= 000012
FF.RWF= 000006          F.FFBY= 000014          GETNAM= ****** GX       L.CNT = 000002      M.PITM= 000000
FF.SPC= 000004          F.FLAG= 000002          GETNB = ****** GX       L.PRNT= 000001      M.PRDS= 000006
FHEADR  005012R 025     F.FLB = 000100          GETNB2= ****** GX       L.PTR = 000000      M.QQAD= 000000
FHRET   005040R 025     F.FNAM= 000110          GETWRD  000144RG 020    L.PTR1= 000004      M.QUAD= 000000
FILNTF= 104467          F.FNB = 000102          GNOME   003102R 025     L.SUBS= 000002      M.RAND= 000012
FIS   = 000000          F.FNDF= 000010          HASHC   005136R 025     MACHIN= 000106      M.RAVL= 000000
FISCON= 046000          F.FTYP= 000116          HASHTB  005140R 025     MAXCHN= 000014      M.RCIP= 000000
FIXPTR  003074R 025     F.FUN = 000010          HSHMSK= 177601          MONBLK  003120R 025 M.RDCT= 000011
FLAGS   004764R 025     F.FVER= 000120          HSHSIZ= 000200          MONCVT  003132R 025 M.REVS= 000003
FLCBUF  003250R 025     F.HEAD= 000006          ICHAN   005050R 025     MONLEN  003130R 025 M.ROLL= 000000
FLINE   005014R 025     F.HIBK= 000004          ILFNAM= 104464          MONPTR  003122R 025 M.SATA= 000001
FLOEXP  003270R 025     F.LBL = 000020          ILLCHN= 104540          MONSFT  003126R 025 M.SCAN= 000011
FLONEG  003272R 025     F.LINE= 000012          ILLSW = 104565          MONTYP  003124R 025 M.SHAP= 000000
FLSWTB  000222R 025     F.LOC = 000004          INDERR= 104410          MONVAL  003134R 025 M.SIGN= 000000
FLTLEN= 000010          F.LSTC= 000004          INNTMP  003250R 025     M.ABS = 000000      M.SUBR= 000001
FNINSI  005044R 025     F.LUN = 000042          INNTM2  005016R 025     M.BOL = 000200      M.SYM = 000203
FORIGN  000032R 025     F.MBCT= 000054          INPTR   000064R 025     M.BXAV= 000007      M.SYN = 000201
FO.APD= 000106          F.MBC1= 000055          INSNAM= 104462          M.BXCR= 000007      M.TRAN= 000012
FO.MFY= 000002          F.MBFG= 000056          IOERR = 104477          M.BXCT= 000007      M.$ABS= 000000
FO.RD = 000001          F.MON = 000002          I.CNT = 177776          M.BXEX= 000007      M.$CEI= 000000
FO.UPD= 000006          F.NIL = 000001          I.DIM = 177772          M.BXFX= 000007      M.$FLR= 000000
FO.WRT= 000016          F.NRBD= 000024          I.LEN = 000012          M.BXIO= 000007      M.$IDN= 000000
FPMODE  000042R 025     F.NREC= 000030          I.OERR= 000062          M.BXLC= 000003      M.$NEG= 000000
FPP   = 000000          F.OPN = 000010          I.OLD = 177770          M.BXNC= 000007      M.$SIG= 000000
FPPLEN= 000004          F.OVBS= 000030          I.PTR = 177774          M.BXNL= 000007      M.1RDT= 000011
FREBLK  007400R 025     F.PREV= 000000          I.STR = 177766          M.BXPP= 000007      M.1REV= 000003
FUZZ    000142R 025     F.PRNT= 100000          LABERR= 104406          M.BXPW= 000007      M.1SCN= 000011
FXPTR1  003076R 025     F.PROT= 010000          LABLPT  005042R 025     M.BXRL= 000007      NAMBUF  003240R 025
F$EVAL= 000100          F.RACC= 000016          LARGBL  003162R 025     M.BXTS= 000007      NB.DEV= 000200
F$EVI = 000200          F.RATT= 000001          LARGCV  003176R 025     M.BXWA= 000007      NB.DIR= 000100
F$EXEC= 000400          F.RCNM= 000034          LARGFL  003160R 025     M.CEI = 177777      NB.NAM= 000004
F$FUNC= 001000          F.RCTL= 000017          LARGLE  003174R 025     M.CHR = 000202      NB.SD1= 000400
F.ACTL= 000076          F.READ= 004000          LARGLN  003166R 025     M.CMT = 000206      NB.SD2= 001000
F.ALOC= 000040          F.REDF= 000040          LARGPT  003164R 025     M.DOUT= 000006      NB.SNM= 000040
F.BBFS= 000062          F.RES = 000040          LARGSF  003172R 025     M.DQAD= 000006      NB.STP= 000020
F.BDB = 000070          F.RET = 177772          LARGTY  003170R 025     M.ENCD= 000004      NB.SVR= 000010
F.BGBC= 000057          F.RSIZ= 000002          LARGVA  003200R 025     M.EXE = 000000      NB.TYP= 000002
F.BITS= 177774          F.RTYP= 000000          LBINC   005100R 025     M.EXP = 000001      NB.VER= 000001
F.BKDN= 000026          F.R2  = 000014          LBUFF   003316R 025     M.FACT= 000001      ND.DEV= 000200
F.BKDS= 000020          F.R3  = 000016          LBUFLN= 000621          M.FIL = 000177      ND.EXT= 000002
F.BKEF= 000050          F.R4  = 000020          LCOUNT  005106R 025     M.FLR = 177777      ND.NAM= 000004
F.BKP1= 000051          F.SEQN= 000100          LENERR= 104412          M.FMT = 000004      ND.PPN= 000100
F.BKST= 000024          F.SPDV= 000072          LENTAB  000054R 025     M.GDUP= 000003      ND.$EX= 000000
F.BKVB= 000064          F.SPUN= 000074          LEXPTR  003100R 025     M.GDWN= 000003      ND.$ES= 000020
F.BR  = 000020          F.STBK= 000036          LFTOP   005066R 025     M.GETI= 000006      ND.$NM= 000000
F.CHR = 000075          F.STK = 000022          LINC    005102R 025     M.IBM = 000003      ND.$NS= 000040
F.CNTG= 000034          F.STOP= 000004          LINE    005016R 025     M.IDN = 000C00      ND.$PN= 001400
F.CODE= 000010          F.STP = 000001          LININC  005026R 025     M.INSP= 000006      NEGSGN  005063R 025
F.DFNB= 000046          F.SUSP= 000002          LISIZE= 000020          M.IOTA= 000000      NEWLIN= ****** GX
F.DSPT= 000044          F.TRC = 000002          LITOP   005072R 025     M.LIT = 000205      NINPT1  003302R 025
F.DVNM= 000134          F.UNIT= 000136          LNLNG = 104541          M.LN  = 000001      NINPT2  003300R 025
F.EFBK= 000010          F.URBD= 000020          LOCLCT  005036R 025     M.MMD = 000010      NINSTK  003304R 025
F.EFN = 000050          F.VAL = 177776          LODOVR= ****** GX       M.NEG = 000000      NINTMP  003310R 025
F.END = 002000          F.VBN = 000064          LOWCOR  000002R 025     M.NOT = 000000      NINTYP  003306R 025
```

SYMBOL TABLE

```
NOTAVL= 104472          OPTMP9  005110R  025 ROVER   003312R  025 SYMTOP  005350R  025 T.DIM = 000100
NOTOPN= 104473          OPTM10  005112R  025 RSTS  = 000000       SYMTP   006272R  025 T.FUN = 010000
NPRT1   004762R  025    OPTM14  005026R  025 RSX   = 000115       SYMWRN= 000054       T.LIST= 000400
NULL    007364R  025    ORIGIN  000030R  025 RTOP    005070R  025 SYNERR= 104407       T.LIT = 000020
NUMCHN= 000020          OUTBLK= ****** GX        RT11  = 000000       SYSERR= 104402       T.MON = 000010
NUMDFL  003230R  025    OUTPTR  000066R  025 R.BITS= 000003       SYSTAT  000000R  024 T.NIL = 000004
NUMEFL  003233R  025    OVROPR= 000000           R.FIX = 000001       SY.BAD= 000001       T.NUM = 002000
NUMESN  003236R  025    OWORD   000126RG 020 R.LCT = 000002       SY.BRK= 100000       T.OPFN= 000001
NUMFLG  003230R  025    PATCH   007344R  025 R.LN  = 000002       SY.BS = 000010       T.OTHR= 040000
NUMLEN  000062R  025    PCOPYC  000176RG 020 R.LNUM= 000010       SY.CLR= 000040       T.SOP = 000002
NUMNEG  003232R  025    PRECSN  000052R  025 R.NEXT= 000006       SY.ERP= 000020       T.SPOP= 000040
NUMQFL  003234R  025    PRINTM= ****** GX        R.OFF = 000000       SY.NOU= 000100       T.TMP = 000200
NUMRIG  003231R  025    RANDAC  003064R  025 R.PREV= 000004       SY.OPN= 000002       T.VAL = 100000
NUMWID  003226R  025    RANDY   003062R  025 R.PTR = 000000       SY.PMT= 001000       T.VAR = 001000
NUMZFL  003235R  025    RANERR= 104411           R.SEQ = 000003       SY.SRT= 000400       UPPER = 177400
NYIERR= 104432          RARGBL  003204R  025 R.VAR = 000002       SY.SVC= 000004       UTNAME  000344R  023
N.BLSZ= 000036          RARGCV  003220R  025 R6    =%000006       SY.SVL= 000200       VALERR= 104413
N.DESC= 000020          RARGFL  003202R  025 R7    =%000007       S.AB  = 177776       VARBLE  007400R  025
N.DEV = 000032          RARGLE  003216R  025 SAVMSG   000000R  002 S.ASB = 100000       VARBLS  000000R  025
N.DEVN= 000034          RARGLN  003210R  025 SAVOVR= ****** GX     S.BFHD= 000020       VERNUM  000000R  025
N.DID = 000024          RARGPT  003206R  025 SI    =%000005       S.BITS= 000002       VRSION= 000002
N.DVNM= 000032          RARGSF  003214R  025 SIDAMN= 104433       S.CHR1= 000003       WIDTH   000072R  025
N.EXT = 000014          RARGTY  003212R  025 SIFRAM  003314R  025 S.FATT= 000016       WRITEB= ****** GX
N.FID = 000000          RARGVA  003222R  025 SISTAK  006350R  025 S.FDB = 000140       WROUND  000020R  025
N.FNAM= 000006          RBASE   005074R  025 SITOP   006346R  025 S.FLB = 040000       WSDATE  000136R  025
N.FTYP= 000014          RCOUNR  005112R  025 SITP    007144R  025 S.FNAM= 000006       WSFULL= 104403
N.FVER= 000016          RCOUNT  005110R  025 SIWARN= 000200       S.FNB = 000036       WSNAME  000074R  025
N.NAME= 000006          READP = ****** GX        SIZTAB  000044R  025 S.FNBW= 000017       WSTIME  000132R  025
N.NEXT= 000022          RED     005070R  025 SI.BIT= 177774       S.FNTY= 000004       $FIS  = ****** G
N.PPN = 000024          REDIG   005066R  025 SI.LEN= 177772       S.FTYP= 000002       $FPP  = ****** G
N.STAT= 000020          REDOFF  005072R  025 SI.RET= 177772       S.LEN = 000002       $FPPLN= ****** G
N.UNIT= 000034          REENTR  003060R  025 SI.VAL= 177776       S.LINK= 177774       $MACHN= ****** G
OBUFLN= 001000          RESBIT  003142R  025 SKPSIZ   005134R  025 S.NFEN= 000020       $RSTS = ****** G
OCHAN   005052R  025    RESCHN= ****** GX        SPHOLD  000004R  024 S.TYPE= 140000       $RSX  = ****** G
OCHAR = ****** GX       RESCVT  003152R  025 SPINIT  000002R  024 S.VAL = 000000       $RT11 = ****** G
OMODE   000402R  023    RESERR= 104440           STKINC= 000774       TKDIMS  005072R  025 .BOOLE= 000002
OPNSET= ****** GX       RESLEN  003150R  025 STOP  = 104420       TKFRAM  005066R  025 .CHARA= 000000
OPTMP0  005066R  025    RESPTR  003140R  025 SUBCNT   003104R  025 TKLEFT  005070R  025 .DEF. = 000002
OPTMP1  005070R  025    RESSFT  003146R  025 SUBDCT   003106R  025 TMODE   000403R  023 .FLOAT= 000006
OPTMP2  005072R  025    RESTYP  003144R  025 SUBPTR   003110R  025 TMPTAG= 000166       .FSRCB= ****** G
OPTMP3  005074R  025    RESULT  003136R  025 SYMBOL= ****** GX     TTYBUF  004140R  025 .INTEG= 000004
OPTMP4  005076R  025    RESVAL  003154R  025 SYMB1    005344R  025 TTYLEN= 000622       .MOLUN  000404R  023
OPTMP5  005100R  025    RINC    005104R  025 SYMF     005340R  025 TURKEY= 104436       .MRKDL= ****** GX
OPTMP6  005102R  025    RLTBAS  005066R  025 SYMPT1   005346R  025 TYPE  = 000166       ...GBL= 000000
OPTMP7  005104R  025    ROTINC  005070R  025 SYMTAB   005352R  025 T.BOL = 020000       ...TPC= 000020
OPTMP8  005106R  025    ROUND   000010R  025 SYMTBF= 104404       T.CON = 004000

. ABS.  000000  000
        000000  001
TABLES  000010  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
```

```
COPYC   MACRO M1113  08-FEB-80 11:24  PAGE 52-4
SYMBOL TABLE

VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    001110      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13080 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:53
COPYC,DM1:[305,301]COPYC/-SP=APLRX1,APLDEF,COPYC,END
```

FILEID**CPIARG

```
  CCCCCCCC   PPPPPPPP      IIIIII      AAAAAA     RRRRRRRR      GGGGGGGG
  CCCCCCCC   PPPPPPPP      IIIIII      AAAAAA     RRRRRRRR      GGGGGGGG
CC           PP      PP      II      AA      AA   RR      RR   GG
CC           PP      PP      II      AA      AA   RR      RR   GG
CC           PP      PP      II      AA      AA   RR      RR   GG
CC           PP      PP      II      AA      AA   RR      RR   GG
CC           PPPPPPPP         II      AA      AA   RRRRRRRR    GG
CC           PPPPPPPP         II      AA      AA   RRRRRRRR    GG
CC           PP               II      AAAAAAAAAA   RR  RR      GG  GGGGGG
CC           PP               II      AAAAAAAAAA   RR  RR      GG  GGGGGG
CC           PP               II      AA      AA   RR    RR    GG      GG      ....
CC           PP               II      AA      AA   RR    RR    GG      GG      ....
  CCCCCCCC   PP            IIIIII     AA      AA   RR      RR    GGGGGG        ....
  CCCCCCCC   PP            IIIIII     AA      AA   RR      RR    GGGGGG        ....


LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:26
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    CPIARG          COPY ARGUMENT FROM THE TEMP FILE
```

CPIARG MACRO M1113 08-FEB-80 11:26 PAGE 1

```
                                
000000                  RT11    =  0            ;NOT RT-11
000000                  RSTS    =  0            ;NOT RSTS/E
000115                  RSX     = 'M            ;RSX-11/M
                        
000000                  EIS     =  0            ;NO EIS
000000                  FIS     =  0            ;NO FIS
000000                  FPP     =  0            ;NO FPP
                        
000004                  FPPLEN  =  4.           ;SINGLE PRECISION
                        
000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CPIARG
                                .IDENT  \V02.03\

                                .SBTTL  CPIARG          COPY ARGUMENT FROM THE TEMP FILE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is used to input elements that have been
                                ;       COPYed from the specified WS into the current WS. The
                                ;       format transmitted is the same as specified in CPOARG.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       The COPYed arguments are placed in the current WS.
                                ;
                                ; ROUTINES CALLED:
                                ;       CPINAM  - Inputs the name of arguments
                                ;       FNHEAD  - Inputs a function header
                                ;       FNLINE  - Inputs a function line
                                ;       GETBLK  - Reads in a block of data
                                ;       GETWRD  - Gets a word of data
                                ;       PRINTM  - Prints a message to the console/execute channel
                                ;--
000000                                  .PSECT  CODE

000000                          CPIARG::
000000                                  CALL    CPINAM                  ;Get the next name
000004  001001                          BNE     1$                      ;   if there is one
000006                                  RETURN                          ;There isn't

000010  010004                  1$:     MOV     R0,R4                   ;Copy the pointer
000012                                  CALL    GETWRD                  ;Read in the # of elements
000016  010003                          MOV     R0,R3                   ;Get that number
000020  001012                          BNE     3$                      ;The number appears to be OK
000022  012701  000000'                 MOV     #OBNTFD,R1              ;'OBJECT NOT FOUND: '
000026                          2$:     CALL    PRINTM                  ;Print the message
000032  010401                          MOV     R4,R1                   ;Copy the name pointer
000034                                  CALL    NAMOUT                  ;Print the name
000040                                  CALL    NEWLIN                  ;   and a <CR><LF>
000044  000755                          BR      CPIARG                  ;Loop for all entries

000046                          3$:     CALL    GLBSCH                  ;Look to see the global
000052  001416                          BEQ     6$                      ;There is none
000054  005737  005070'                 TST     OPTMP1                  ;Is the command ')PCOPY'?
000060  001022                          BNE     7$                      ;Yes, don't copy the object
000062  013701  003314'                 MOV     SIFRAM,R1               ;Get the pointer to the SI Frame
000066  032761  001000  000002  4$:     BIT     #F$FUNC,F.FLAG(R1)      ;Is it a Function Execution Frame?
000074  001403                          BEQ     5$                      ;No, ignore this frame
000076  021061  000010                  CMP     (R0),F.FUN(R1)          ;Is it the same as the object?
000102  001411                          BEQ     7$                      ;Yes, we're not allowed to copy it
000104  011101                  5$:     MOV     (R1),R1                 ;Go up a level on the SI
000106  001767                          BEQ     4$                      ;All finished? Maybe more stuff
000110                          6$:     PUSH    <R4,(R0)>               ;Save the name pointer, value direct
000114  020327  000001                  CMP     R3,#1.                  ;What type of data is it?
000120  001443                          BEQ     16$                     ;Array Block
000122  002460                          BLT     17$                     ;List
000124  000510                          BR      20$                     ;Function Definition
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 45
CPIARG          COPY ARGUMENT FROM THE TEMP FILE


                                        ;+
                                        ; SPECIAL CASE - LET'S SKIP THIS GARBAGE
                                        ;-
000126  005703                  7$:    TST     R3              ;Take absolute value of #
000130  003002                         BGT     8$              ;It's non-negative, is OK
000132  001432                         BEQ     14$             ;Zero - skip the skipping
000134  005403                         NEG     R3              ;Negate the count
000136                           8$:    CALL    GETWRD          ;Get the # of words to skip
000142  010001                         MOV     R0,R1           ;Save the count
000144  001404                         BEQ     10$             ;Nothing there
000146                           9$:    CALL    GETBYT          ;Get a byte
000152                                  SOB     R1,9$           ;Loop for all of it
000156                           10$:   SOB     R3,8$           ;Loop for each element
000162  000416                         BR      14$             ;Now, we print error msg
                                        ;+
                                        ; FINISH UP THE ARGUMENT:
                                        ;-
000164  010600                  11$:   MOV     SP,R0           ;Pointer ** 2 to Old
000166                                  CALL    ERASE           ;Eliminate old value
000172  001010                         BNE     13$             ;Erase attempt failed
000174  005726                         TST     (SP)+           ;Pop the name
000176  010436                         MOV     R4,@(SP)+       ;New value is ...
000200  000677                         BR      CPIARG          ;   and loop for each name

000202  010416                  12$:   MOV     R4,(SP)         ;Store the allocated stuff
000204  010600                         MOV     SP,R0           ;Point to it
000206                                  CALL    ERASE           ;Get rid of it
000212  001005                         BNE     15$             ;The attempt failed, bad error
000214  005726                  13$:   TST     (SP)+           ;pop this stuff
000216                                  POP     R4              ;Get name pointer
000220  012701  000024'         14$:   MOV     #OBNTCP,R1      ;'OBJECT NOT COPIED: ''
000224  000700                         BR      2$              ;Print the message

000226  104402                  15$:   SYSERR                  ;An error
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 46
CPIARG          COPY ARGUMENT FROM THE TEMP FILE


                                        ;+
                                        ; Copy in an Array Block
                                        ;-
000230                          16$:    CALL    GETWRD          ;Get the total size of the item
000234                                  PUSH    R0              ;Save it for transfer
000236                                  CALL    GETCO1          ;Do a conditional Memory Request
000242  000757                          BR      12$             ;Error, say not copied
000244  011604                          MOV     (SP),R4         ;Restore the length
000246  010001                          MOV     R0,R1           ;Copy result field pointer
000250  010016                          MOV     R0,(SP)         ;Save the new value's address
000252                                  CALL    GETBLK          ;Read in the data
000256                                  POP     R4              ;Restore the address
000260  060414                          ADD     R4,(R4)         ;Point to self
000262  000740                          BR      11$             ;   and finish up


                                        ;+
                                        ; Here, we input list blocks
                                        ;-
000264  005403                  17$:    NEG     R3              ;Make the count positive
000266  010300                          MOV     R3,R0           ;Copy the count
000270  006300                          ASL     R0              ;Convert to bytes
000272  062700  000006                  ADD     #L.PTR1+2.,R0   ;Add in overhead
000276                                  CALL    GETCO1          ;Attempt to allocate space
000302  000744                          BR      13$             ;The attempt failed
000304  010002                          MOV     R0,R2           ;Copy the result pointer
000306                                  PUSH    R0              ;Save it on the stack, too
000310  010212                          MOV     R2,(R2)         ;Point to self
000312  062722  000006                  ADD     #L.PTR1+2.,(R2)+ ;Add in offset to pointer
000316  110322                          MOVB    R3,(R2)+        ;# of elements
000320  105022                          CLRB    (R2)+           ;No flag bits
000322  011022                          MOV     (R0),(R2)+      ;Pointer to last element
000324  005703                          TST     R3              ;Any elements at all?
000326  001405                          BEQ     19$             ;No
000330                          18$:    CALL    CPINAM          ;Input a name
000334  010022                          MOV     R0,(R2)+        ;Point to the name entry
000336                                  SOB     R3,18$          ;Loop for each entry
000342                          19$:    POP     R4              ;Restore pointer to block
000344  000707                          BR      11$             ;   and go
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 47
CPIARG          COPY ARGUMENT FROM THE TEMP FILE


                                        ;+
                                        ; This section of code handles function definition
                                        ;-
                                        ;
000346                          20$:    PUSH    (R4)                    ;Save the old value
000350  005014                          CLR     (R4)                    ;Now clear it for later
000352  005303                          DEC     R3                      ;Skip the header
000354  010337  005072'                 MOV     R3,OPTMP2               ;Save the count
000360  005037  005044'                 CLR     FNINSI                  ;No, it is OK to do this
000364  005037  005012'                 CLR     FHEADR                  ;No function yet
000370                                  CALL    GETWRD                  ;Get the length
000374  010004                          MOV     R0,R4                   ;Copy for block read
000376                                  PUSH    R0                      ;Save size of line
000400  012701  000000G                 MOV     #CPOUTL,R1              ;Temporary space for copy
000404                                  CALL    GETBLK                  ;Read in a block of info
000410  012700  000000G                 MOV     #CPOUTL,R0              ;Point to the buffer
000414                                  POP     R1                      ;Restore count
000416                                  CALL    FNHEAD                  ;Get the function header
000422  104402                          SYSERR                          ;Should not ever happen
000424  010037  005012'                 MOV     R0,FHEADR               ;Point to the function header
000430  012676  000002                  MOV     (SP)+,@2.(SP)           ;Restore local value
000434  005037  005016'                 CLR     LINE                    ;Header is line 0
000440  005237  005016'         21$:    INC     LINE                    ;Start w/ line 1
000444                                  CALL    GETWRD                  ;Read in the next line's length
000450  010004                          MOV     R0,R4                   ;Copy for block read
000452  001415                          BEQ     22$                     ;Nothing to read
000454                                  PUSH    R0                      ;Save for later
000456  012701  000000G                 MOV     #CPOUTL,R1              ;Tempory space for COPY
000462                                  CALL    GETBLK                  ;Read in the stuff
000466  012704  000000G                 MOV     #CPOUTL,R4              ;Pointer to buffer
000472                                  POP     R3                      ;Count remaining
000474                                  CALL    GETNB                   ;Scan into the line
000500                                  CALL    FNLINE                  ;Enter a line
000504  104402                          SYSERR                          ;Should never happen
000506  005337  005072'         22$:    DEC     OPTMP2                  ;Decrement count
000512  003352                          BGT     21$                     ;   and loop for all
000514                                  CALL    FNEND                   ;Take care of labels, etc.
000520  000621                          BR      11$                     ;All is ok, continue


000000                                  .PSECT  TABLES
000000                          OBNTFD:
000000      157     142     152         .ASCIZ  /object not found>  /
000003      145     143     164
000006      040     156     157
000011      164     040     146
000014      157     165     156
000017      144     076     040
000022      040     000


000024                          OBNTCP:
000024      157     142     152         .ASCIZ  /object not copied> /
000027      145     143     164
000032      040     156     157
000035      164     040     143
000040      157     160     151
000043      145     144     076
000046      040     000
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 47-1
CPIARG          COPY ARGUMENT FROM THE TEMP FILE

    180                                         .EVEN
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 48
CPIARG          COPY ARGUMENT FROM THE TEMP FILE

     1          000001                          .END
```

CPIARG MACRO M1113 08-FEB-80 11:26 PAGE 48-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    CTRMAX  000172R  025 D.CTLM= 000000    EDIFLG  005062R  025
AC1    =%000001          CHRLPR= 000072    C.ALOC= 000162       D.DEAL= 000003    EDIT   = 000003
AC2    =%000002          CHRMNS= 000137    C.BLOK= 000064       D.DECD= 000002    EIS    = 000000
AC3    =%000003          CHRMUL= 000075    C.BLSZ= 000140       D.DIV = 000000    ELESIZ  005132R  025
AC4    =%000004          CHRNEG= 000100    C.BUFF= 177760       D.DMD = 000010    ENDCOR  000004R  025
AC5    =%000005          CHRPDL= 000271    C.BUFL= 177762       D.DQAD= 177777    EOF    = 104475
ALLINT  003274R  025     CHRPLU= 000055    C.CCNT= 177766       D.DROP= 000002    ERASE  = ****** GX
ARYRET  003276R  025     CHRPWR= 000120    C.CPTR= 177764       D.ENCD= 000002    ERRMAX= 000166
ASCQUO= 000042           CHRQST= 000121    C.CZ   = 001000      D.EQ   = 000000    ERRNUM  004772R  025
A.ALEN= 000004           CHRQUO= 000113    C.DIG  = 000040      D.EXPD= 000002    ERRPTR  004770R  025
A.ARG  = 000040          CHRRBR= 000047    C.FASW= 020000       D.FGET= 000005    ERRO    004774R  025
A.ASB  = 000200          CHRRPR= 000042    C.FLEN= 000010       D.FMT  = 000004    EXEBUF  005054R  025
A.BITS= 000003           CHRSEM= 000074    C.FLSH= 004000       D.FPNT= 000005    E$CT    005056R  025
A.CON  = 000020          CHRSMO= 000112    C.FRC  = 000001      D.FPUT= 000005    E$PT    005060R  025
A.DIMS= 000002           CHRSPA= 000040    C.IDEN= 000010       D.GE   = 000000    FA.APD= 000100
A.DIM1= 000006           CHRUPA= 000131    C.NAME= 000102       D.GT   = 000000    FA.CRE= 000010
A.NUM  = 000006          CHR.BS= 000010    C.NOBF= 000400       D.IBM  = 000002    FA.DLK= 001000
A.PTR  = 000000          CHR.CC= 000003    C.NUM  = 000020      D.IND  = 000002    FA.ENB= 100000
A.TMP  = 000010          CHR.CR= 000015    C.OP   = 000004      D.INNR= 000013    FA.EXC= 002000
BADDIR= 104463           CHR.CZ= 000032    C.OPN  = 010000      D.INTR= 000006    FA.EXT= 000004
BADERR= 104437           CHR.GT= 000046    C.RDLN= 177776       D.LE   = 000000    FA.NSP= 000100
BLZBUB  005116R  025     CHR.LF= 000012    C.REWT= 002000       D.LOG  = 000001    FA.NSR= 000200
BRSELF  005046R  025     CHR.LT= 000043    C.RND  = 000010      D.LT   = 000000    FA.POS= 010000
CCNT    005072R  025     CHR.SI= 000017    C.SBIT= 177756       D.MAX  = 000000    FA.RD  = 000001
CHANEX  003060R  025     CHR.SL= 000057    C.SPFX= 000100       D.MEM  = 000002    FA.RWD= 004000
CHAN00  000256R  025     CHR.SO= 000016    C.SPGL= 000200       D.MIN  = 000000    FA.SEQ= 040000
CHAN01  000330R  025     CHR.TB= 000011    C.SPLX= 000001       D.MUL  = 000000    FA.SHR= 000040
CHAN02  000512R  025     CHR.UL= 000106    C.SPRS= 000002       D.NE   = 000000    FA.TMP= 000020
CHAN03  000674R  025     CHR..A= 000141    C.SRC  = 177774      D.NN   = 000000    FA.WCK= 020000
CHAN04  001056R  025     CHR..E= 000145    C.SRCP= 000024       D.NR   = 000000    FA.WRT= 000002
CHAN05  001240R  025     CHR..S= 000163    C.TT   = 000004      D.OR   = 000000    FD.BLK= 000010
CHAN06  001422R  025     CHR..T= 000164    C.WIDC= 177770       D.PLUS= 000000    FD.CCL= 000002
CHAN07  001604R  025     CHR..U= 000165    C.WIDR= 177772       D.PSUB= 000006    FD.COM= 020000
CHAN08  001766R  025     CHR..Z= 000172    DEBUG  = 000000      D.PWR  = 000001    FD.CR  = 000002
CHAN09  002150R  025     CHR..0= 000060    DEFERR= 104405       D.QQAD= 000000    FD.DIR= 000010
CHAN10  002332R  025     CHR..1= 000061    DEVIL   005114R  025 D.QUAD= 000000    FD.FTN= 000001
CHAN11  002514R  025     CH.AND= 000001    DEVNTF= 104470       D.RES  = 000000    FD.F11= 040000
CHAN12  002676R  025     CMDBUF  003250R  025 DIGITS  000006R  025 D.RHO  = 000000    FD.INS= 000010
CHAN13  000022R  023     CMDMAX= 000010    DIMAX  = 000020      D.ROT  = 000003    FD.ISP= 002000
CHAN14  000204R  023     CNTSIZ  005130R  025 DIMBLK  003112R  025 D.SCON= 000006    FD.MNT= 100000
CHNOPN= 104471           COMBIN  005076R  025 DIMPTR  003114R  025 D.SDIF= 000006    FD.OSP= 004000
CHRAND= 000051           COMCNT  005126R  025 DIMTYP  003116R  025 D.SUB  = 000000    FD.PLC= 000004
CHRBEL= 000007           COMERR= 104425       DIRFUL= 104466       D.SUBS= 000006    FD.PRN= 000004
CHRBKR= 000133           COMPTR  005122R  025 DOMERR= 104417       D.TAKE= 000002    FD.PSE= 010000
CHRBOL= 000200           COMRTN  003156R  025 DPTERR= 104416       D.TRAN= 000004    FD.RAH= 000001
CHRBOX= 000114           COMSKP  005124R  025 DYDCNT  003227R  025 D.UNON= 000006    FD.RAN= 000002
CHRCMA= 000054           CORPTR  005120R  025 D.AND  = 000000      D.$MAX= 000000    FD.REC= 000001
CHRCOL= 000076           CPIARG  000000RG  020 D.BXCT= 000007      D.$MIN= 000000    FD.RWM= 000001
CHRDEL= 000107           CPINAM= ****** GX     D.BXIO= 000007      D.$MUL= 000000    FD.SDI= 000020
CHRDIV= 000053           CPOUTL= ****** GX     D.BXNL= 000007      D.$PLU= 000000    FD.SQD= 000040
CHRDLR= 000176           CS.ASS= 000200        D.BXPP= 000007      D.$RES= 000000    FD.TTY= 000004
CHRDLT= 000110           CS.BOO= 000010        D.BXPW= 000007      D.$SUB= 000000    FD.WBH= 000002
CHRDOT= 000056           CS.FLO= 000020        D.BXRL= 000007      D.1COM= 000001    FF.CHR= 000005
CHREP  = 000105          CS.IDN= 000040        D.CIRC= 000001      D.1EXP= 000002    FF.NV  = 000003
CHRERR= 104431           CS.ONE= 000100        D.COMB= 000001      D.1ROT= 000003    FF.POE= 000002
CHRFIL= 000177           CTEPS   000152R  025 D.COMP= 000001      EDICHR  005063R  025 FF.RWD= 000001
CHRFRO= 000135           CTMAX   000212R  025 D.CONT= 000006      EDICNT  005064R  025 FF.RWF= 000006
```

SYMBOL TABLE

```
FF.SPC= 000004          F.ERR = 000052          GETBLK= ****** GX       L.BITS= 000003          M.OUTR= 000012
FHEADR  005012R   025  F.FACC= 000043          GETBYT= ****** GX       L.CNT = 000002          M.PITM= 000000
FHRET   005040R   025  F.FFBY= 000014          GETCO1= ****** GX       L.PRNT= 000001          M.PRDS= 000006
FILNTF= 104467          F.FLAG= 000002          GETNB = ****** GX       L.PTR = 000000          M.QQAD= 000000
FIS   = 000000          F.FLB = 000100          GETWRD= ****** GX       L.PTR1= 000004          M.QUAD= 000000
FISCON= 046000          F.FNAM= 000110          GLBSCH= ****** GX       L.SUBS= 000002          M.RAND= 000012
FIXPTR  003074R   025  F.FNB = 000102          GNOME   003102R   025  MACHIN= 000106          M.RAVL= 000000
FLAGS   004764R   025  F.FNDF= 000010          HASHC   005136R   025  MAXCHN= 000014          M.RCIP= 000000
FLCBUF  003250R   025  F.FTYP= 000116          HASHTB  005140R   025  MONBLK  003120R   025  M.RDCT= 000011
FLINE   005014R   025  F.FUN = 000010          HSHMSK= 177601          MONCVT  003132R   025  M.REVS= 000003
FLOEXP  003270R   025  F.FVER= 000120          HSHSIZ= 000200          MONLEN  003130R   025  M.ROLL= 000000
FLONEG  003272R   025  F.HEAD= 000006          ICHAN   005050R   025  MONPTR  003122R   025  M.SATA= 000001
FLSWTB  000222R   025  F.HIBK= 000004          ILFNAM= 104464          MONSFT  003126R   025  M.SCAN= 000011
FLTLEN= 000010          F.LBL = 000020          ILLCHN= 104540          MONTYP  003124R   025  M.SHAP= 000000
FNEND = ****** GX       F.LINE= 000012          ILLSW = 104565          MONVAL  003134R   025  M.SIGN= 000000
FNHEAD= ****** GX       F.LOC = 000004          INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FNINSI  005044R   025  F.LSTC= 000004          INNTMP  003250R   025  M.BOL = 000200          M.SYM = 000203
FNLINE= ****** GX       F.LUN = 000042          INNTM2  005016R   025  M.BXAV= 000007          M.SYN = 000201
FORIGN  000032R   025  F.MBCT= 000054          INPTR   000064R   025  M.BXCR= 000007          M.TRAN= 000012
FO.APD= 000106          F.MBC1= 000055          INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FO.MFY= 000002          F.MBFG= 000056          IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FO.RD = 000001          F.MON = 000002          I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FO.UPD= 000006          F.NIL = 000001          I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FO.WRT= 000016          F.NRBD= 000024          I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FPMODE  000042R   025  F.NREC= 000030          I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FPP   = 000000          F.OPN = 000010          I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FPPLEN= 000004          F.OVBS= 000030          I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FREBLK  007400R   025  F.PREV= 000000          I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FUZZ    000142R   025  F.PRMT= 100000          LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R   025
FXPTR1  003076R   025  F.PROT= 010000          LABLPT  005042R   025  M.BXTS= 000007          NAMOUT= ****** GX
F$EVAL= 000100          F.RACC= 000016          LARGBL  003162R   025  M.BXWA= 000007          NB.DEV= 000200
F$EVI = 000200          F.RATT= 000001          LARGCV  003176R   025  M.CEI = 177777          NB.DIR= 000100
F$EXEC= 000400          F.RCNM= 000034          LARGFL  003160R   025  M.CHR = 000202          NB.NAM= 000004
F$FUNC= 001000          F.RCTL= 000017          LARGLE  003174R   025  M.CMT = 000206          NB.SD1= 000400
F.ACTL= 000076          F.READ= 004000          LARGLN  003166R   025  M.DOUT= 000006          NB.SD2= 001000
F.ALOC= 000040          F.REDF= 000040          LARGPT  003164R   025  M.DQAD= 000006          NB.SNM= 000040
F.BBFS= 000062          F.RES = 000040          LARGSF  003172R   025  M.ENCD= 000004          NB.STP= 000020
F.BDB = 000070          F.RET = 177772          LARGTY  003170R   025  M.EXE = 000000          NB.SVR= 000010
F.BGBC= 000057          F.RSIZ= 000002          LARGVA  003200R   025  M.EXP = 000001          NB.TYP= 000002
F.BITS= 177774          F.RTYP= 000000          LBINC   005100R   025  M.FACT= 000001          NB.VER= 000001
F.BKDN= 000026          F.R2  = 000014          LBUFF   003316R   025  M.FIL = 000177          ND.DEV= 000200
F.BKDS= 000020          F.R3  = 000016          LBUFLN= 000621          M.FLR = 177777          ND.EXT= 000002
F.BKEF= 000050          F.R4  = 000020          LCOUNT  005106R   025  M.FMT = 000004          ND.NAM= 000004
F.BKP1= 000051          F.SEQN= 000100          LENERR= 104412          M.GDUP= 000003          ND.PPN= 000100
F.BKST= 000024          F.SPDV= 000072          LENTAB  000054R   025  M.GDWN= 000003          ND.$EX= 000000
F.BKVB= 000064          F.SPUN= 000074          LEXPTR  003100R   025  M.GETI= 000006          ND.$E$= 000020
F.BR  = 000020          F.STBK= 000036          LFTOP   005066R   025  M.IBM = 000003          ND.$NM= 000000
F.CHR = 000075          F.STK = 000022          LINC    005102R   025  M.IDN = 000000          ND.$N$= 000040
F.CNTG= 000034          F.STOP= 000004          LINE    005016R   025  M.INSP= 000006          ND.$PN= 001400
F.CODE= 000010          F.STP = 000001          LININC  005026R   025  M.IOTA= 000000          NEGSGN  005063R   025
F.DFNB= 000046          F.SUSP= 000002          LISIZE= 000020          M.LIT = 000205          NEWLIN= ****** GX
F.DSPT= 000044          F.TRC = 000002          LITOP   005072R   025  M.LN  = 000001          NINPT1  003302R   025
F.DVNM= 000134          F.UNIT= 000136          LNLNG = 104541          M.MMD = 000010          NINPT2  003300R   025
F.EFBK= 000010          F.URBD= 000020          LOCLCT  005036R   025  M.NEG = 000000          NINSTK  003304R   025
F.EFN = 000050          F.VAL = 177776          LOWCOR  000002R   025  M.NOT = 000000          NINTMP  003310R   025
F.END = 002000          F.VBN = 000064          LSTERP  004766R   025  M.NUB = 000006          NINTYP  003306R   025
F.EOBB= 000032          F.VBSZ= 000060          LWIDTH  000070R   025  M.NUM = 000204          NOTAVL= 104472
```

SYMBOL TABLE

```
NOTOPN= 104473          OPTMP8  005106R     025 RSX    = 000115          SYNERR= 104407          T.DIM = 000100
NPRT1   004762R     025 OPTMP9  005110R     025 RTOP     005070R     025 SYSERR= 104402          T.FUN = 010000
NULL    007364R     025 OPTM10  005112R     025 RT11   = 000000          SYSTAT  000000R     024 T.LIST= 000400
NUMCHN= 000020          OPTM14  005026R     025 R.BITS= 000003          SY.BAD= 000001          T.LIT = 000020
NUMDFL  003230R     025 ORIGIN  000030R     025 R.FIX  = 000001          SY.BRK= 100000          T.MON = 000010
NUMEFL  003233R     025 OUTPTR  000066R     025 R.LCT  = 000002          SY.BS = 000010          T.NIL = 000004
NUMESN  003236R     025 OVROPR= 000000          R.LN   = 000002          SY.CLR= 000040          T.NUM = 002000
NUMFLG  003230R     025 PATCH   007344R     025 R.LNUM= 000010          SY.ERP= 000020          T.OPFN= 000001
NUMLEN  000062R     025 PRECSN  000052R     025 R.NEXT= 000006          SY.NOU= 000100          T.OTHR= 040000
NUMNEG  003232R     025 PRINTM= ****** GX       R.OFF  = 000000          SY.OPN= 000002          T.SOP = 000002
NUMQFL  003234R     025 RANDAC  003064R     025 R.PREV= 000004          SY.PMT= 001000          T.SPOP= 000040
NUMRIG  003231R     025 RANDY   003062R     025 R.PTR  = 000000          SY.SRT= 000400          T.TMP = 000200
NUMWID  003226R     025 RANERR= 104411          R.SEQ  = 000003          SY.SVC= 000004          T.VAL = 100000
NUMZFL  003235R     025 RARGBL  003204R     025 R.VAR  = 000002          SY.SVL= 000200          T.VAR = 001000
NYIERR= 104432          RARGCV  003220R     025 R6     =%000006          S.AB   = 177776          UPPER = 177400
N.BLSZ= 000036          RARGFL  003202R     025 R7     =%000007          S.ASB  = 100000          UTNAME  000344R     023
N.DESC= 000020          RARGLE  003216R     025 SI     =%000005          S.BFHD= 000020          VALERR= 104413
N.DEV = 000032          RARGLN  003210R     025 SIDAMN= 104433          S.BITS= 000002          VARBLE  007400R     025
N.DEVN= 000034          RARGPT  003206R     025 SIFRAM  003314R     025 S.CHR1= 000003          VARBLS  000000R     025
N.DID = 000024          RARGSF  003214R     025 SISTAK  006350R     025 S.FATT= 000016          VERNUM  000000R     025
N.DVNM= 000032          RARGTY  003212R     025 SITOP   006346R     025 S.FDB  = 000140          VERSION= 000002
N.EXT = 000014          RARGVA  003222R     025 SITP    007144R     025 S.FLB  = 040000          WIDTH   000072R     025
N.FID = 000000          RBASE   005074R     025 SIWARN= 000200          S.FNAM= 000006          WROUND  000020R     025
N.FNAM= 000006          RCOUNR  005112R     025 SIZTAB  000044R     025 S.FNB  = 000036          WSDATE  000136R     025
N.FTYP= 000014          RCOUNT  005110R     025 SI.BIT= 177774          S.FNBW= 000017          WSFULL= 104403
N.FVER= 000016          REDCNT  005070R     025 SI.LEN= 177772          S.FNTY= 000004          WSNAME  000074R     025
N.NAME= 000006          REDIGN  005066R     025 SI.RET= 177772          S.FTYP= 000002          WSTIME  000132R     025
N.NEXT= 000022          REDOFF  005072R     025 SI.VAL= 177776          S.LEN  = 000002          $FIS   = ****** G
N.PPN = 000024          REENTR  003060R     025 SKPSIZ  005134R     025 S.LINK= 177774          $FPP   = ****** G
N.STAT= 000020          RESBIT  003142R     025 SPHOLD  000004R     024 S.NFEN= 000020          $FPPLN= ****** G
N.UNIT= 000034          RESCVT  003152R     025 SPINIT  000002R     024 S.TYPE= 140000          $MACHN= ****** G
OBNTCP  000024R     002 RESERR= 104440          STKINC= 000774          S.VAL  = 000000          $RSTS  = ****** G
OBNTFD  000000R     002 RESLEN  003150R     025 STOP   = 104420          TKDIMS  005072R     025 $RSX   = ****** G
OBUFLN= 001000          RESPTR  003140R     025 SUBCNT  003104R     025 TKFRAM  005066R     025 $RT11  = ****** G
OCHAN   005052R     025 RESSFT  003146R     025 SUBDCT  003106R     025 TKLEFT  005070R     025 .BOOLE= 000002
OMODE   000402R     023 RESTYP  003144R     025 SUBPTR  003110R     025 TMODE   000403R     023 .CHARA= 000000
OPTMP0  005066R     025 RESULT  003136R     025 SYMB1   005344R     025 TMPTAG= 000166          .DEF.  = 000002
OPTMP1  005070R     025 RESVAL  003154R     025 SYMF    005340R     025 TTYBUF  004140R     025 .FLOAT= 000006
OPTMP2  005072R     025 RINC    005104R     025 SYMPT1  005346R     025 TTYLEN= 000622          .FSRCB= ****** G
OPTMP3  005074R     025 RLTBAS  005066R     025 SYMTAB  005352R     025 TURKEY= 104436          .INTEG= 000004
OPTMP4  005076R     025 ROTINC  005070R     025 SYMTBF= 104404          TYPE   = 000166          .MOLUN  000404R     023
OPTMP5  005100R     025 ROUND   000010R     025 SYMTOP  005350R     025 T.BOL  = 020000          ...GBL= 000000
OPTMP6  005102R     025 ROVER   003312R     025 SYMTP   006272R     025 T.CON  = 004000          ...TPC= 000020
OPTMP7  005104R     025 RSTS   = 000000          SYMWRN= 000054
```

```
. ABS.  000000      000
        000000      001
TABLES  000050      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   0000C0      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
```

```
CPIARG  MACRO M1113  08-FEB-80 11:26  PAGE 48-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000522      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13120 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:49
CPIARG,DM1:[305,301]CPIARG/-SP=APLRX1,APLDEF,CPIARG,END
```

```
**FILE**ID**CPOARG
```

```
  CCCCCCCC    PPPPPPPP       000000       AAAAAA     RRRRRRRR       GGGGGGGG
  CCCCCCCC    PPPPPPPP       000000       AAAAAA     RRRRRRRR       GGGGGGGG
CC            PP      PP   00      00    AA    AA   RR      RR     GG
CC            PP      PP   00      00    AA    AA   RR      RR     GG
CC            PP      PP   00      00    AA    AA   RR      RR     GG
CC            PP      PP   00      00    AA    AA   RR      RR     GG
CC            PPPPPPPP     00      00    AA    AA   RRRRRRRR       GG
CC            PPPPPPPP     00      00    AA    AA   RRRRRRRR       GG
CC            PP           00      00    AAAAAAAAAA   RR  RR         GG  GGGGGG
CC            PP           00      00    AAAAAAAAAA   RR  RR         GG  GGGGGG
CC            PP           00      00    AA    AA   RR    RR       GG      GG      ....
CC            PP           00      00    AA    AA   RR    RR       GG      GG      ....
  CCCCCCCC    PP             000000      AA    AA   RR      RR       GGGGGG        ....
  CCCCCCCC    PP             000000      AA    AA   RR      RR       GGGGGG        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:28
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    CPOARG           COPY ARGUMENT TO TMP FILE
```

CPOARG MACRO M1113 08-FEB-80 11:28
TABLE OF CONTENTS

46- 66 CPOELE COPY OUT AN ELEMENT

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 1


      1        000000                  RT11    =  0            ;NOT RT-11
      2        000000                  RSTS    =  0            ;NOT RSTS/E
      3        000115                  RSX     = 'M            ;RSX-11/M
      4
      5        000000                  EIS     =  0            ;NO EIS
      6        000000                  FIS     =  0            ;NO FIS
      7        000000                  FPP     =  0            ;NO FPP
      8
      9        000004                  FPPLEN  =  4.           ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CPOARG
                                .IDENT  \V02.03\

                                .SBTTL  CPOARG          COPY ARGUMENT TO TMP FILE

                                ;++
                                ; FUNCTIONAL SPECIFICATION:
                                ;       This routine is used to elements (Arrays, Functions,
                                ;       and Groups) to the temporary file for later input
                                ;       and addition to the COPYing workspace. The format
                                ;       for the file is:
                                ;
                                ;           1) Name     - The Name in Symbol Table Format
                                ;           2) Count    - The number of items/type specifier
                                ;                         for Array blocks, = 1
                                ;                         for Functions,    = n (Positive Integer > 1)
                                ;                         for Groups,       =-n (Negative Integer)
                                ;                         for Nothing,      = 0
                                ;
                                ;       For Array Blocks:
                                ;           3) Array    - The Array Block in internal format, save the
                                ;                         1st word which is an offset to the data. (i.e.,
                                ;                         you add the Array Block's new address to the
                                ;                         first word in the array block to set it up)
                                ;
                                ;       For Functions: (For each of n times, n = # of lines in the function)
                                ;           3) Function - The function line, in FIXLIN format (Size
                                ;                         rounded up to words, pad character as needed)
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       The data specified by the Symbol table Pointer is output to
                                ;       the file.
                                ;
                                ; ROUTINES CALLED:
                                ;       FLNLST  - List a line of the function
                                ;       OUTBLK  - Output a block of data to the TMP file
                                ;       OWORD   - Output a word of data
                                ;--
000000                          .PSECT  CODE
                                .ENABL  LSB
```

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 45
CPOARG           COPY ARGUMENT TO TMP FILE

     42 000000                          CPOARG::
     43                                 ;+
     44                                 ; Now we read back in the list of names, and write out the items
     45                                 ;-
     46 000000  012704  005344'                 MOV     #SYMB1,R4               ;Point to start of alphabetic list
     47 000004                                  CALL    GETWRD                  ;Read in # of elements to read
     48 000010  010002                          MOV     R0,R2                   ;Copy the count
     49 000012  001415                          BEQ     3$                      ;Nothing to copy
     50 000014                          1$:     CALL    CPINAM                  ;Copy in the name
     51 000020  010004                          MOV     R0,R4                   ;Move ptr to R4
     52 000022                                  CALL    CPOELE                  ;Output the argument
     53 000026                                  SOB     R2,1$                   ;Loop for each element
     54 000032  000410                          BR      4$                      ;Finish up
     55                                 ;
     56                                 ; Special Case - we write out everything
     57                                 ;
     58 000034                          2$:     CALL    GLBSCH                  ;Find global
     59 000040  001402                          BEQ     3$                      ;Nothing there
     60 000042                                  CALL    CPOELE                  ;Output the arg
     61 000046  016404  177776          3$:     MOV     S.AB(R4),R4             ;Pointer to next element
     62 000052  001370                          BNE     2$                      ;Loop for all
     63 000054  005000                  4$:     CLR     R0                      ;Clear R0
     64 000056                                  CALLR   OWORD                   ;Output end flag, quit
```

```
                                        .SBTTL  CPOELE          COPY OUT AN ELEMENT

                                        ;++
                                        ; Functional Description:
                                        ;       This routine copys a specified item to the current
                                        ;       output channel in ")COPY" format
                                        ;
                                        ; Inputs:
                                        ;       R4      - The Pointer to the item in the Symbol Table
                                        ;--
000062                                          .PSECT  CODE

000062                                  CPOELE: PUSH    <R1,R2,R3,R4>           ;Save the name pointer
000072                                          CALL    GLBSCH                  ;Do a search for the global value
000076  001403                                  BEQ     6$                      ;No, finished
000100  021027  177777                          CMP     (R0),#-1.               ;Already been copied?
000104  001475                                  BEQ     13$                     ;Yes, finished
000106                                  6$:     PUSH    R0                      ;Save data pointer
000110  010401                                  MOV     R4,R1                   ;Copy the symbol table entry
000112                                          CALL    CPONAM                  ;Output the name
000116                                          POP     R0                      ;Point to it
000120  011001                                  MOV     (R0),R1                 ;Copy pointer to item
000122  001463                                  BEQ     12$                     ;Nothing there, error
000124  012710  177777                          MOV     #-1.,(R0)               ;    but don't let it happen again!
000130  116104  000003                          MOVB    A.BITS(R1),R4           ;Get the data type
000134                                          ASH     #-5.,R4                 ;Shift High bits to 2X position
000150  042704  177771                          BIC     #^C<6>,R4               ;Clear out unwanted bits
000154  060407                                  ADD     R4,PC                   ;'CASE' Statement

000156  000403                                  BR      7$                      ;Groups (& Lists)
000160  000530                                  BR      15$                     ;Functions
000162  000453                                  BR      14$                     ;Arrays
000164  104402                                  SYSERR                          ;SYSTEM ERROR

000166                                  7$:     CALL    18$                     ;Output the names of the elements
000172  116103  000002                          MOVB    L.CNT(R1),R3            ;Get the Loop Count
000176  001440                                  BEQ     13$                     ;Nothing there, finished
000200  011101                                  MOV     (R1),R1                 ;Point to 1st element
000202  012104                          8$:     MOV     (R1)+,R4                ;Get pointer to item
000204                                          PUSH    R1                      ;Save R1 for Later
000206                                          CALL    GLBSCH                  ;Do a search for the global value
000212  001423                                  BEQ     10$                     ;Nothing there, pretend
000214  011001                                  MOV     (R0),R1                 ;Copy the pointer
000216  020127  177777                          CMP     R1,#-1.                 ;Has it already been COPYed?
000222  001417                                  BEQ     10$                     ;Yes, don't do it again
000224  032761  140000  000002                  BIT     #<S.ASB+S.FLB>,2.(R1)   ;Is it a group?
000232  001010                                  BNE     9$                      ;No, so recurse
000234                                          PUSH    R1                      ;Save Pointer to Item
000236  010401                                  MOV     R4,R1                   ;Copy pointer to name
000240                                          CALL    CPONAM                  ;Output the name
000244                                          POP     R1                      ;Restore the data pointer
000246                                          CALL    18$                     ;Output the list
000252  000403                                  BR      10$                     ;    but not each item

000254  010400                          9$:     MOV     R4,R0                   ;Copy the symbol table pointer
000256                                          CALL    CPOELE                  ;A little recursion, if you please
000262                                  10$:    POP     R1                      ;Restore R1
```

```
    123 000264                                  SOB     R3,8$                   ;Loop for all the elements
    124 000270  000403                          BR      13$                     ;Fini'
    125
    126 000272  005000              12$:        CLR     R0                      ;Clear the count
    127 000274                                  CALL    OWORD                   ;Output zero length
    128 000300              13$:                POP     <R4,R3,R2,R1>           ;Restore the name pointer
    129 000310                                  RETURN                          ;Finished
```

```
                                        ;+
                                        ; Output an Array
                                        ;-
000312  116104  000003        14$:    MOVB    A.BITS(R1),R4           ;Get the type
000316  132704  000020            BITB    #A.CON,R4               ;Is it a label?
000322  001363                     BNE     12$                     ;Yes, must not copy it
000324  012700  000001            MOV     #1.,R0                  ;Indicate 1 Block, is Array
000330                             CALL    OWORD                   ;Output word count
000334                             PUSH    <R1,(R1)>               ;Save pointers
000340  042704  177771            BIC     #^C<A.NUM>,R4           ;Extract it from BITS field
000344  016100  000004            MOV     A.ALEN(R1),R0           ;Get the # of elements
000350                             ASH     SIZTAB(R4),R0           ;Get the size in bytes
000364  005200                     INC     R0                      ;Round to
000366  042700  000001            BIC     #<1>,R0                 ;   words
000372                             PUSH    R0                      ;Save count for data
000374  116104  000002            MOVB    A.DIMS(R1),R4           ;Get dim count
000400                             CONINC  R4                      ;(Cheat for scalar)
000404  006304                     ASL     R4                      ;   times 2
000406  062704  000006            ADD     #A.DIM1,R4              ;Plus overhead
000412  060400                     ADD     R4,R0                   ;This is the total size, in bytes
000414                             CALL    OWORD                   ;Output the Total Size
000420  010411                     MOV     R4,(R1)                 ;Move it to the offset field
000422                             CALL    OUTBLK                  ;Output the block
000426                             POP     <R4,R1>                 ;Restore some pointers
000432  010136                     MOV     R1,@(SP)+               ;Also an indirect
000434                             CALL    OUTBLK                  ;Output the data
000440  000717                     BR      13$                     ;And go
```

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 48
CPOELE           COPY OUT AN ELEMENT

    159                                 ;+
    160                                 ; Output a Function
    161                                 ;-
    162 000442  010137  005012'     15$:    MOV     R1,FHEADR           ;Set a pointer to the Function Header
    163 000446  005000                      CLR     R0                  ;Start w/ none
    164 000450  016101  000006      16$:    MOV     R.NEXT(R1),R1       ;Point to the next line
    165 000454  005200                      INC     R0                  ;Increment line count
    166 000456  020137  005012'             CMP     R1,FHEADR           ;Finished yet?
    167 000462  001372                      BNE     16$                 ;Nope, keep looping
    168 000464  010003                      MOV     R0,R3               ;Copy the real count
    169 000466  020027  000001              CMP     R0,#1.              ;Only 1 element?
    170 000472                              CONINC  R0                  ;Cheat if only 1
    171 000476                              CALL    OWORD               ;Output the count
    172 000502                      17$:    PUSH    R.NEXT(R1)          ;Save the pointer to the next line
    173 000506  012700  003036'             MOV     #C.SBIT+CHANEX,R0   ;Point to the output channel
    174 000512  012720  010400              MOV     #C.OPN!C.NOBF,(R0)+ ;Channel is open, special buffering
    175 000516  012720  000000G             MOV     #CPOUTL,(R0)+       ;Output to copy utility area
    176 000522  012720  001000              MOV     #512.,(R0)+         ;Size of Util area
    177 000526  012720  000000G             MOV     #CPOUTL,(R0)+       ;Start at the beginning
    178 000532  012720  001000              MOV     #512.,(R0)+         ;Adjust accordingly
    179 000536  005020                      CLR     (R0)+               ;No width
    180 000540  005020                      CLR     (R0)+               ;    at all
    181 000542                              PUSH    OCHAN               ;Save output channel
    182 000546  012737  003060' 005052'     MOV     #CHANEX,OCHAN       ;Output channel
    183 000554                              CALL    FLNLST              ;List the line to Utility Area
    184 000560                              POP     OCHAN               ;Restore output channel
    185 000564  013704  003044'             MOV     C.CPTR+CHANEX,R4    ;Pointer past last char used
    186 000570  112724  000040              MOVB    #CHRSPA,(R4)+       ;Pad whethter we  need it or not
    187 000574  042704  000001              BIC     #<1>,R4             ;Even # of words
    188 000600  013701  003040'             MOV     CHANEX+C.BUFF,R1    ;Point to the Buffer
    189 000604  160104                      SUB     R1,R4               ;Subtract to get size
    190 000606  010400                      MOV     R4,R0               ;Copy the size
    191 000610                              CALL    OWORD               ;Output size, in bytes
    192 000614                              CALL    OUTBLK              ;Output it
    193 000620                              POP     R1                  ;Restore pointer to the next line
    194 000622                              SOB     R3,17$              ;Loop for each line
    195 000626  020161  000006              CMP     R1,R.NEXT(R1)       ;Only 1 Line?
    196 000632  001222                      BNE     13$                 ;Nope
    197 000634  000616                      BR      12$                 ;Finished
```

```
                                        ;+
                                        ; Copy a group to the file
                                        ;-
000636                          18$:    PUSH    <R1,R3>             ;Save the pointer to the data
000642  116100  000002                  MOVB    L.CNT(R1),R0        ;Get # of elements in the group
000646  010003                          MOV     R0,R3               ;Save count for loop
000650  005400                          NEG     R0                  ;Hack for List
000652                                  CALL    OWORD               ;Output the # of Elements
000656  001406                          BEQ     20$                 ;If nothing, do nothing
000660  011104                          MOV     (R1),R4             ;Pointer to the list
000662  012401                  19$:    MOV     (R4)+,R1            ;Point to the 1st name
000664                                  CALL    CPONAM              ;Output a name
000670                                  SOB     R3,19$              ;Loop for all the names
000674                          20$:    POP     <R3,R1>             ;Restore Pointer
000700                                  RETURN                      ;FINISHED

                                        .DSABL  LSB
```

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 50
CPOELE           COPY OUT AN ELEMENT

      1          000001                              .END
```

CPOARG MACRO M1113 08-FEB-80 11:28 PAGE 50-1
SYMBOL TABLE

```
AC     =%000000          CHRLBR= 000073    CTEPS   000152R 025 D.COMP= 000001    EDICHR  005063R 025
AC1    =%000001          CHRLPR= 000072    CTMAX   000212R 025 D.CONT= 000006    EDICNT  005064R 025
AC2    =%000002          CHRMNS= 000137    CTRMAX  000172R 025 D.CTLM= 000000    EDIFLG  005062R 025
AC3    =%000003          CHRMUL= 000075    C.ALOC= 000162      D.DEAL= 000003    EDIT   = 000003
AC4    =%000004          CHRNEG= 000100    C.BLOK= 000064      D.DECD= 000002    EIS    = 000000
AC5    =%000005          CHRPDL= 000271    C.BLSZ= 000140      D.DIV = 000000    ELESIZ  005132R 025
ALLINT  003274R 025      CHRPLU= 000055    C.BUFF= 177760      D.DMD = 000010    ENDCOR  000004R 025
ARYRET  003276R 025      CHRPWR= 000120    C.BUFL= 177762      D.DQAD= 177777    EOF    = 104475
ASCQUO= 000042           CHRQST= 000121    C.CCNT= 177766      D.DROP= 000002    ERRMAX= 000166
A.ALEN= 000004           CHRQUO= 000113    C.CPTR= 177764      D.ENCD= 000002    ERRNUM  004772R 025
A.ARG = 000040           CHRRBR= 000047    C.CZ   = 001000     D.EQ   = 000000    ERRPTR  004770R 025
A.ASB = 000200           CHRRPR= 000042    C.DIG = 000040      D.EXPD= 000002    ERR0    004774R 025
A.BITS= 000003           CHRSEM= 000074    C.FASW= 020000      D.FGET= 000005    EXEBUF  005054R 025
A.CON = 000020           CHRSMO= 000112    C.FLEN= 000010      D.FMT = 000004    E$CT    005056R 025
A.DIMS= 000002           CHRSPA= 000040    C.FLSH= 004000      D.FPNT= 000005    E$PT    005060R 025
A.DIM1= 000006           CHRUPA= 000131    C.FRC = 000001      D.FPUT= 000005    FA.APD= 000100
A.NUM = 000006           CHR.BS= 000010    C.IDEN= 000010      D.GE   = 000000    FA.CRE= 000010
A.PTR = 000000           CHR.CC= 000003    C.NAME= 000102      D.GT   = 000000    FA.DLK= 001000
A.TMP = 000010           CHR.CR= 000015    C.NOBF= 000400      D.IBM = 000002    FA.ENB= 100000
BADDIR= 104463           CHR.CZ= 000032    C.NUM = 000020      D.IND = 000002    FA.EXC= 002000
BADERR= 104437           CHR.GT= 000046    C.OP   = 000004     D.INNR= 000013    FA.EXT= 000004
BLZBUB  005116R 025      CHR.LF= 000012    C.OPN = 010000      D.INTR= 000006    FA.NSP= 000100
BRSELF  005046R 025      CHR.LT= 000043    C.RDLN= 177776      D.LE   = 000000    FA.NSR= 000200
CCNT    005072R 025      CHR.SI= 000017    C.REWT= 002000      D.LOG = 000001    FA.POS= 010000
CHANEX  003060R 025      CHR.SL= 000057    C.RND = 000010      D.LT   = 000000    FA.RD = 000001
CHAN00  000256R 025      CHR.SO= 000016    C.SBIT= 177756      D.MAX = 000000    FA.RWD= 004000
CHAN01  000330R 025      CHR.TB= 000011    C.SPFX= 000100      D.MEM = 000002    FA.SEQ= 040000
CHAN02  000512R 025      CHR.UL= 000106    C.SPGL= 000200      D.MIN = 000000    FA.SHR= 000040
CHAN03  000674R 025      CHR..A= 000141    C.SPLX= 000001      D.MUL = 000000    FA.TMP= 000020
CHAN04  001056R 025      CHR..E= 000145    C.SPRS= 000002      D.NE   = 000000    FA.WCK= 020000
CHAN05  001240R 025      CHR..S= 000163    C.SRC  = 177774     D.NN   = 000000    FA.WRT= 000002
CHAN06  001422R 025      CHR..T= 000164    C.SRCP= 000024      D.NR   = 000000    FD.BLK= 000010
CHAN07  001604R 025      CHR..U= 000165    C.TT   = 000004     D.OR   = 000000    FD.CCL= 000002
CHAN08  001766R 025      CHR..Z= 000172    C.WIDC= 177770      D.PLUS= 000000    FD.COM= 020000
CHAN09  002150R 025      CHR..0= 000060    C.WIDR= 177772      D.PSUB= 000006    FD.CR  = 000002
CHAN10  002332R 025      CHR..1= 000061    DEBUG  = 000000     D.PWR  = 000001    FD.DIR= 000010
CHAN11  002514R 025      CH.AND= 000001    DEFERR= 104405      D.QQAD= 000000    FD.FTN= 000001
CHAN12  002676R 025      CMDBUF  003250R 025 DEVIL   005114R 025 D.QUAD= 000000    FD.F11= 040000
CHAN13  000022R 023      CMDMAX= 000010    DEVNTF= 104470      D.RES  = 000000    FD.INS= 000010
CHAN14  000204R 023      CNTSIZ  005130R 025 DIGITS  000006R 025 D.RHO  = 000000    FD.ISP= 002000
CHNOPN= 104471           COMBIN  005076R 025 DIMAX  = 000020     D.ROT  = 000003    FD.MNT= 100000
CHRAND= 000051           COMCNT  005126R 025 DIMBLK  003112R 025 D.SCON= 000006    FD.OSP= 004000
CHRBEL= 000007           COMERR= 104425      DIMPTR  003114R 025 D.SDIF= 000006    FD.PLC= 000004
CHRBKR= 000133           COMPTR  005122R 025 DIMTYP  003116R 025 D.SUB  = 000000    FD.PRN= 000004
CHRBOL= 000200           COMRTN  003156R 025 DIRFUL= 104466      D.SUBS= 000006    FD.PSE= 010000
CHRBOX= 000114           COMSKP  005124R 025 DOMERR= 104417      D.TAKE= 000002    FD.RAH= 000001
CHRCMA= 000054           CORPTR  005120R 025 DPTERR= 104416      D.TRAN= 000004    FD.RAN= 000002
CHRCOL= 000076           CPINAM= ****** GX   DYDCNT  003224R 025 D.UNON= 000006    FD.REC= 000001
CHRDEL= 000107           CPOARG  000000RG 020 D.AND  = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRDIV= 000053           CPOELE  000062R  020 D.BXCT= 000007     D.$MIN= 000000    FD.SDI= 000020
CHRDLR= 000176           CPONAM= ****** GX    D.BXIO= 000007     D.$MUL= 000000    FD.SQD= 000040
CHRDLT= 000110           CPOUTL= ****** GX    D.BXNL= 000007     D.$PLU= 000000    FD.TTY= 000004
CHRDOT= 000056           CS.ASS= 000200       D.BXPP= 000007     D.$RES= 000000    FD.WBH= 000002
CHREP  = 000105          CS.BOO= 000010       D.BXPW= 000007     D.$SUB= 000000    FF.CHR= 000005
CHRERR= 104431           CS.FLO= 000020       D.BXRL= 000007     D.1COM= 000001    FF.NV  = 000003
CHRFIL= 000177           CS.IDN= 000040       D.CIRC= 000001     D.1EXP= 000002    FF.POE= 000002
CHRFRO= 000135           CS.ONE= 000100       D.COMB= 000001     D.1ROT= 000003    FF.RWD  000001
```

CPOARG MACRO M1113 08-FEB-80 11:28 PAGE 50-2
SYMBOL TABLE

```
FF.RWF= 000006          F.FACC= 000043
FF.SPC= 000004          F.FFBY= 000014
FHEADR  005012R     025 F.FLAG= 000002
FHRET   005040R     025 F.FLB = 000100
FILNTF= 104467          F.FNAM= 000110
FIS   = 000000          F.FNB = 000102
FISCON= 046000          F.FNDF= 000010
FIXPTR  003074R     025 F.FTYP= 000116
FLAGS   004764R     025 F.FUN = 000010
FLCBUF  003250R     025 F.FVER= 000120
FLINE   005014R     025 F.HEAD= 000006
FLNLST= ****** GX       F.HIBK= 000004
FLOEXP  003270R     025 F.LBL = 000020
FLONEG  003272R     025 F.LINE= 000012
FLSWTB  000222R     025 F.LOC = 000004
FLTLEN= 000010          F.LSTC= 000004
FNINSI  005044R     025 F.LUN = 000042
FORIGN  000032R     025 F.MBCT= 000054
FO.APD= 000106          F.MBC1= 000055
FO.MFY= 000002          F.MBFG= 000056
FO.RD = 000001          F.MON = 000002
FO.UPD= 000006          F.NIL = 000001
FO.WRT= 000016          F.NRBD= 000024
FPMODE  000042R     025 F.NREC= 000030
FPP   = 000000          F.OPN = 000010
FPPLEN= 000004          F.OVBS= 000030
FREBLK  007400R     025 F.PREV= 000000
FUZZ    000142R     025 F.PRNT= 100000
FXPTR1  003076R     025 F.PROT= 010000
FSEVAL= 000100          F.RACC= 000016
FSEVI = 000200          F.RATT= 000001
FSEXEC= 000400          F.RCNM= 000034
FSFUNC= 001000          F.RCTL= 000017
F.ACTL= 000076          F.READ= 004000
F.ALOC= 000040          F.REDF= 000040
F.BBFS= 000062          F.RES = 000040
F.BDB = 000070          F.RET = 177772
F.BGBC= 000057          F.RSIZ= 000002
F.BITS= 177774          F.RTYP= 000000
F.BKDN= 000026          F.R2  = 000014
F.BKDS= 000020          F.R3  = 000016
F.BKEF= 000050          F.R4  = 000020
F.BKP1= 000051          F.SEQN= 000100
F.BKST= 000024          F.SPDV= 000072
F.BKVB= 000064          F.SPUN= 000074
F.BR  = 000020          F.STBK= 000036
F.CHR = 000075          F.STK = 000022
F.CNTG= 000034          F.STOP= 000004
F.CODE= 000010          F.STP = 000001
F.DFNB= 000046          F.SUSP= 000002
F.DSPT= 000044          F.TRC = 000002
F.DVNM= 000134          F.UNIT= 000136
F.EFBK= 000010          F.URBD= 000020
F.EFN = 000050          F.VAL = 177776
F.END = 002000          F.VBN = 000064
F.EOBB= 000032          F.VBSZ= 000060
F.ERR = 000052          GETWRD= ****** GX

GLBSCH= ****** GX       L.PTR1= 000004          M.QUAD= 000000
GNOME   003102R     025 L.SUBS= 000002          M.RAND= 000012
HASHC   005136R     025 MACHIN= 000106          M.RAVL= 000000
HASHTB  005140R     025 MAXCHN= 000014          M.RCIP= 000000
HSHMSK= 177601          MONBLK  003120R     025 M.RDCT= 000011
HSHSIZ= 000200          MONCVT  003132R     025 M.REVS= 000003
IASH  = ****** GX       MONLEN  003130R     025 M.ROLL= 000000
ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
LINC    005[illegible]  025 M.IDN = 000000          ND.$PN= 001400
LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
LITOP   005[illegible]R 025 M.LN  = 000001          NINSTK  003304R     025
LNLNG = 10454[illegible]    M.MMD = 000010          NINTMP  003310R     025
LOCLCT  005036[illegible]   025 M.NEG = 000000          NINTYP  003306R     025
LOWCOR  000002R     025 M.NOT = 000000          NOTAVL= 104472
LSTERP  004766R     025 M.NUB = 000006          NOTOPN= 104473
LWIDTH  000070R     025 M.NUM = 000204          NPRT1   004762R     025
L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
```

CPOARG MACRO M1113 08-FEB-80 11:28 PAGE 50-3
SYMBOL TABLE

```
NUMESN  003236R     025 OVROPR= 000000        R.FIX  = 000001        SYSTAT  000000R     024 T.FUN  = 010000
NUMFLG  003230R     025 OWORD  = ****** GX    R.LCT  = 000002        SY.BAD= 000001         T.LIST= 000400
NUMLEN  000062R     025 PATCH    007344R      025 R.LN   = 000002    SY.BRK= 100000         T.LIT  = 000020
NUMNEG  003232R     025 PRECSN   000052R      025 R.LNUM= 000010     SY.BS  = 000010         T.MON  = 000010
NUMQFL  003234R     025 RANDAC   003064R      025 R.NEXT= 000006     SY.CLR= 000040         T.NIL  = 000004
NUMRIG  003231R     025 RANDY    003062R      025 R.OFF  = 000000    SY.ERP= 000020         T.NUM  = 002000
NUMWID  003226R     025 RANERR= 104411        R.PREV= 000004         SY.NOU= 000100         T.OPFN= 000001
NUMZFL  003235R     025 RARGBL   003204R      025 R.PTR  = 000000    SY.OPN= 000002         T.OTHR= 040000
NYIERR= 104432          RARGCV   003220R      025 R.SEQ  = 000003    SY.PMT= 001000         T.SOP  = 000002
N.BLSZ= 000036          RARGFL   003202R      025 R.VAR  = 000002    SY.SRT= 000400         T.SPOP= 000040
N.DESC= 000020          RARGLE   003216R      025 R6     =%000006    SY.SVC= 000004         T.TMP  = 000200
N.DEV  = 000032         RARGLN   003210R      025 R7     =%000007    SY.SVL= 000200         T.VAL  = 100000
N.DEVN= 000034          RARGPT   003206R      025 SI     =%000005    S.AB   = 177776         T.VAR  = 001000
N.DID  = 000024         RARGSF   003214R      025 SIDAMN= 104433     S.ASB  = 100000         UPPER  = 177400
N.DVNM= 000032          RARGTY   003212R      025 SIFRAM  003314R  025 S.BFHD= 000020         UTNAME  000344R     023
N.EXT  = 000014         RARGVA   003222R      025 SISTAK  006350R  025 S.BITS= 000002         VALERR= 104413
N.FID  = 000000         RBASE    005074R      025 SITOP   006346R  025 S.CHR1= 000003         VARBLE  007400R     025
N.FNAM= 000006          RCOUNR   005112R      025 SITP    007144R  025 S.FATT= 000016         VARBLS  000000R     025
N.FTYP= 000014          RCOUNT   005110R      025 SIWARN= 000200     S.FDB  = 000140         VERNUM  000000R     025
N.FVER= 000016          REDCNT   005070R      025 SIZTAB  000044R  025 S.FLB  = 040000         VRSION= 000002
N.NAME= 000006          REDIGN   005066R      025 SI.BIT= 177774     S.FNAM= 000006         WIDTH   000072R     025
N.NEXT= 000022          REDOFF   005072R      025 SI.LEN= 177772     S.FNB  = 000036         WROUND  000020R     025
N.PPN  = 000024         REENTR   003060R      025 SI.RET= 177772     S.FNBW= 000017         WSDATE  000136R     025
N.STAT= 000020          REGTYP= 000000        SI.VAL= 177776         S.FNTY= 000004         WSFULL= 104403
N.UNIT= 000034          RESBIT   003142R      025 SKPSIZ  005134R  025 S.FTYP= 000002         WSNAME  000074R     025
OBUFLN= 001000          RESCVT   003152R      025 SPHOLD  000004R  024 S.LEN  = 000002         WSTIME  000132R     025
OCHAN   005052R     025 RESERR= 104440        SPINIT  000002R  024 S.LINK= 177774         $FIS   = ****** G
OMODE   000402R     023 RESLEN   003150R      025 STKINC= 000774     S.NFEN= 000020         $FPP   = ****** G
OPTMP0  005066R     025 RESPTR   003140R      025 STOP   = 104420    S.TYPE= 140000         $FPPLN= ****** G
OPTMP1  005070R     025 RESSFT   003146R      025 SUBCNT  003104R  025 S.VAL  = 000000         $MACHN= ****** G
OPTMP2  005072R     025 RESTYP   003144R      025 SUBDCT  003106R  025 TKDIMS  005072R     025 $RSTS  = ****** G
OPTMP3  005074R     025 RESULT   003136R      025 SUBPTR  003110R  025 TKFRAM  005066R     025 $RSX   = ****** G
OPTMP4  005076R     025 RESVAL   003154R      025 SYMB1   005344R  025 TKLEFT  005070R     025 $RT11  = ****** G
OPTMP5  005100R     025 RINC     005104R      025 SYMF    005340R  025 TMODE   000403R     023 .BOOLE= 000002
OPTMP6  005102R     025 RLTBAS   005066R      025 SYMPT1  005346R  025 TMPTAG= 000166          .CHARA= 000000
OPTMP7  005104R     025 ROTINC   005070R      025 SYMTAB  005352R  025 TTYBUF  004140R     025 .DEF.  = 000002
OPTMP8  005106R     025 ROUND    000010R      025 SYMTBF= 104404     TTYLEN= 000622         .FLOAT= 000006
OPTMP9  005110R     025 ROVER    003312R      025 SYMTOP  005350R  025 TURKEY= 104436         .FSRCB= ****** G
OPTM10  005112R     025 RSTS   = 000000        SYMTP   006272R  025 TYPE   = 000166         .INTEG= 000004
OPTM14  005026R     025 RSX    = 000115        SYMWRN= 000054         T.BOL  = 020000         .MOLUN  000404R     023
ORIGIN  000030R     025 RTOP     005070R      025 SYNERR= 104407     T.CON  = 004000         ...GBL= 000000
OUTBLK= ****** GX       RT11   = 000000        SYSERR= 104402         T.DIM  = 000100         ...TPC= 000020
OUTPTR  000066R     025 R.BITS= 000003

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
```

```
CPOARG  MACRO M1113  08-FEB-80 11:28  PAGE 50-4
SYMBOL TABLE

TRNTBL  000000     015
MNTBL   000000     016
MNTRTL  000000     017
CODE    000702     020
PATCH   000000     021
$$FSR1  000000     022
DEF$0   000406     023
DEF$1   000006     024
DEF$2   007400     025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13044 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:40
CPOARG,DM1:[305,301]CPOARG/-SP=APLRX1,APLDEF,CPOARG,END
```

```
**FILE**ID**CRTFIL
```

```
  CCCCCCCC   RRRRRRRR     TTTTTTTTTT   FFFFFFFFFF     IIIIII     LL
  CCCCCCCC   RRRRRRRR     TTTTTTTTTT   FFFFFFFFFF     IIIIII     LL
CC           RR      RR       TT       FF             II       LL
CC           RR      RR       TT       FF             II       LL
CC           RR      RR       TT       FF             II       LL
CC           RR      RR       TT       FF             II       LL
CC           RRRRRRRR         TT       FFFFFFFF       II       LL
CC           RRRRRRRR         TT       FFFFFFFF       II       LL
CC           RR  RR           TT       FF             II       LL
CC           RR  RR           TT       FF             II       LL
CC           RR    RR         TT       FF             II       LL                 ....
CC           RR    RR         TT       FF             II       LL                 ....
  CCCCCCCC   RR      RR       TT       FF         IIIIIII      LLLLLLLLLL         ....
  CCCCCCCC   RR      RR       TT       FF         IIIIIII      LLLLLLLLLL         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:30
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    COMMANDS                 CREATE
```

```
CRTFIL  MACRO M1113  08-FEB-80 11:30  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  CRTFIL
                                .IDENT  \V02.03\
                                .SBTTL  COMMANDS                CREATE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       CREATE A FILE BUT DO NOT ATTACH IT TO A PARTICULAR CHANNEL
                                ;--
000000                                  .PSECT  CODE

000000                          CRTFIL::
000000                                  PUSH    <R2,R3,R4>      ;SAVE INPUT REGS
000006  122702  000040          10$:    CMPB    #CHRSPA,R2      ;SPACE?
000012  001004                          BNE     11$             ;NO
000014                                  GETC    13$             ;GET NEXT CHAR
000014  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000016  002421                          BLT 13$                 ;NO, BRANCH TO 13$
000020  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000022  000771                          BR      10$             ;IGNORING SPACES

000024  010200                  11$:    MOV     R2,R0           ;COPY CURRENT CHAR
000026                                  POP     <R4,R3,R2>      ;RESTORE INPUT REGS
000034  132760  000040  000000G       BITB    #C.DIG,CHRTAB(R0);IS IT A DIGIT?
000042  001407                          BEQ     13$             ;NO, SO NO CHANNEL SPECIFIED
000044                                  CALL    CMDCHN          ;GET FILE CHANNEL TO USE
000050  032762  010000  177756          BIT     #C.OPN,C.SBIT(R2) ;CHANNEL OPEN?
000056  001403                          BEQ     14$             ;No, is Alright
000060  104471                          CHNOPN                  ;Yes, An Error

000062  012702  000204'          13$:    MOV     #CHAN14,R2      ;USE UTILITY CHANNEL
000066                          14$:    CALL    RESCHN          ;Clear the channel, ust in case
000072  010200                          MOV     R2,R0           ;Point to FDB
000074  010201                          MOV     R2,R1           ;Point to the
000076  062701  000102                  ADD     #C.NAME,R1      ;    Name Block
000102  012702  000000G                 MOV     #DFLTEX,R2      ;Default name block
000106                                  CALL    DECNAM          ;Get the file name
000112  012701  177777                  MOV     #-1,R1          ;SET 'CREATE' FLAG
000116                                  CALL    OPNSET          ;OPEN THE FILE
000122  022702  000204'                 CMP     #CHAN14,R2      ;USING UTILITY CHANNEL?
000126  001002                          BNE     2$              ;IF NOT, KEEP FILE OPEN ON CHANNEL
000130                          1$:     CALL    RESCHN          ;AND CLOSE IT
000134                          2$:     RETURN                  ;DONE!
```

```
CRTFIL  MACRO M1113  08-FEB-80 11:30  PAGE 46
COMMANDS                CREATE

      1          000001                              .END
```

CRTFIL MACRO M1113 08-FEB-80 11:30 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073          CTRMAX  000172R     025 D.COMP= 000001          EDICHR  005063R     025
AC1    =%000001          CHRLPR= 000072          C.ALOC= 000162          D.CONT= 000006          EDICNT  005064R     025
AC2    =%000002          CHRMNS= 000137          C.BLOK= 000064          D.CTLM= 000000          EDIFLG  005062R     025
AC3    =%000003          CHRMUL= 000075          C.BLSZ= 000140          D.DEAL= 000003          EDIT   = 000003
AC4    =%000004          CHRNEG= 000100          C.BUFF= 177760          D.DECD= 000002          EIS    = 000000
AC5    =%000005          CHRPDL= 000271          C.BUFL= 177762          D.DIV = 000000          ELESIZ  005132R     025
ALLINT  003274R     025  CHRPLU= 000055          C.CCNT= 177766          D.DMD = 000010          ENDCOR  000004R     025
ARYRET  003276R     025  CHRPWR= 000120          C.CPTR= 177764          D.DQAD= 177777          EOF    = 104475
ASCQUO= 000042           CHRQST= 000121          C.CZ   = 001000         D.DROP= 000002          ERRMAX= 000166
A.ALEN= 000004           CHRQUO= 000113          C.DIG  = 000040         D.ENCD= 000002          ERRNUM  004772R     025
A.ARG  = 000040          CHRRBR= 000047          C.FASW= 020000          D.EQ   = 000000         ERRPTR  004770R     025
A.ASB  = 000200          CHRRPR= 000042          C.FLEN= 000010          D.EXPD= 000002          ERRO    004774R     025
A.BITS= 000003           CHRSEM= 000074          C.FLSH= 004000          D.FGET= 000005          EXEBUF  005054R     025
A.CON  = 000020          CHRSMO= 000112          C.FRC  = 000001         D.FMT  = 000004         E$CT    005056R     025
A.DIMS= 000002           CHRSPA= 000040          C.IDEN= 000010          D.FPNT= 000005          E$PT    005060R     025
A.DIM1= 000006           CHRTAB= ****** GX       C.NAME= 000102          D.FPUT= 000005          FA.APD= 000100
A.NUM  = 000006          CHRUPA= 000131          C.NOBF= 000400          D.GE   = 000000         FA.CRE= 000010
A.PTR  = 000000          CHR.BS= 000010          C.NUM  = 000020         D.GT   = 000000         FA.DLK= 001000
A.TMP  = 000010          CHR.CC= 000003          C.OP   = 000004         D.IBM  = 000002         FA.ENB= 100000
BADDIR= 104463           CHR.CR= 000015          C.OPN  = 010000         D.IND  = 000002         FA.EXC= 002000
BADERR= 104437           CHR.CZ= 000032          C.RDLN= 177776          D.INNR= 000013          FA.EXT= 000004
BLZBUB  005116R     025  CHR.GT= 000046          C.REWT= 002000          D.INTR= 000006          FA.NSP= 000100
BRSELF  005046R     025  CHR.LF= 000012          C.RND  = 000010         D.LE   = 000000         FA.NSR= 000200
CCNT    005072R     025  CHR.LT= 000043          C.SBIT= 177756          D.LOG  = 000001         FA.POS= 010000
CHANEX  003060R     025  CHR.SI= 000017          C.SPFX= 000100          D.LT   = 000000         FA.RD  = 000001
CHAN00  000256R     025  CHR.SL= 000057          C.SPGL= 000200          D.MAX  = 000000         FA.RWD= 004000
CHAN01  000330R     025  CHR.SO= 000016          C.SPLX= 000001          D.MEM  = 000002         FA.SEQ= 040000
CHAN02  000512R     025  CHR.TB= 000011          C.SPRS= 000002          D.MIN  = 000000         FA.SHR= 000040
CHAN03  000674R     025  CHR.UL= 000106          C.SRC  = 177774         D.MUL  = 000000         FA.TMP= 000020
CHAN04  001056R     025  CHR..A= 000141          C.SRCP= 000024          D.NE   = 000000         FA.WCK= 020000
CHAN05  001240R     025  CHR..E= 000145          C.TT   = 000004         D.NN   = 000000         FA.WRT= 000002
CHAN06  001422R     025  CHR..S= 000163          C.WIDC= 177770          D.NR   = 000000         FD.BLK= 000010
CHAN07  001604R     025  CHR..T= 000164          C.WIDR= 177772          D.OR   = 000000         FD.CCL= 000002
CHAN08  001766R     025  CHR..U= 000165          DEBUG  = 000000         D.PLUS= 000000          FD.COM= 020000
CHAN09  002150R     025  CHR..Z= 000172          DECNAM= ****** GX       D.PSUB= 000006          FD.CR  = 000002
CHAN10  002332R     025  CHR..0= 000060          DEFERR= 104405          D.PWR  = 000001         FD.DIR= 000010
CHAN11  002514R     025  CHR..1= 000061          DEVIL   005114R     025 D.QQAD= 000000          FD.FTN= 000001
CHAN12  002676R     025  CH.AND= 000001          DEVNTF= 104470          D.QUAD= 000000          FD.F11= 040000
CHAN13  000022R     023  CMDBUF  003250R     025 DFLTEX= ****** GX       D.RES  = 000000         FD.INS= 000010
CHAN14  000204R     023  CMDCHN= ****** GX       DIGITS  000006R     025 D.RHO  = 000000         FD.ISP= 002000
CHNOPN= 104471           CMDMAX= 000010          DIMAX  = 000020         D.ROT  = 000003         FD.MNT= 100000
CHRAND= 000051           CNTSIZ  005130R     025 DIMBLK  003112R     025 D.SCON= 000006          FD.OSP= 004000
CHRBEL= 000007           COMBIN  005076R     025 DIMPTR  003114R     025 D.SDIF= 000006          FD.PLC= 000004
CHRBKR= 000133           COMCNT  005126R     025 DIMTYP  003116R     025 D.SUB  = 000000         FD.PRN= 000004
CHRBOL= 000200           COMERR= 104425          DIRFUL= 104466          D.SUBS= 000006          FD.PSE= 010000
CHRBOX= 000114           COMPTR  005122R     025 DOMERR= 104417          D.TAKE= 000002          FD.RAH= 000001
CHRCMA= 000054           COMRTN  003156R     025 DPTERR= 104416          D.TRAN= 000004          FD.RAN= 000002
CHRCOL= 000076           COMSKP  005124R     025 DYDCNT  003224R     025 D.UNON= 000006          FD.REC= 000001
CHRDEL= 000107           CORPTR  005120R     025 D.AND  = 000000         D.$MAX= 000000          FD.RWM= 000001
CHRDIV= 000053           CRTFIL  000000RG    020 D.BXCT= 000007          D.$MIN= 000000          FD.SDI= 000020
CHRDLR= 000176           CS.ASS= 000200          D.BXIO= 000007          D.$MUL= 000000          FD.SQD= 000040
CHRDLT= 000110           CS.BOO= 000010          D.BXNL= 000007          D.$PLU= 000000          FD.TTY= 000004
CHRDOT= 000056           CS.FLO= 000020          D.BXPP= 000007          D.$RES= 000000          FD.WBH= 000002
CHREP  = 000105          CS.IDN= 000040          D.BXPW= 000007          D.$SUB= 000000          FF.CHR= 000005
CHRERR= 104431           CS.ONE= 000100          D.BXRL= 000007          D.1COM= 000001          FF.NV  = 000003
CHRFIL= 000177           CTEPS   000152R     025 D.CIRC= 000001          D.1EXP= 000002          FF.POE= 000002
CHRFRO= 000135           CTMAX   000212R     025 D.COMB= 000001          D.1ROT= 000003          FF.RWD= 000001
```

SYMBOL TABLE

```
FF.RWF= 000006          F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011
FF.SPC= 000004          F.FLAG= 000002          HSHMSK= 177601      MONCVT  003132R 025 M.REVS= 000003
FHEADR  005012R 025     F.FLB = 000100          HSHSIZ= 000200      MONLEN  003130R 025 M.ROLL= 000000
FHRET   005040R 025     F.FNAM= 000110          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025     F.FUN = 000010          INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025     F.FVER= 000120          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025     F.HEAD= 000006          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025     F.HIBK= 000004          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025     F.LBL = 000020          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025     F.LINE= 000012          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025     F.LOC = 000004          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025     F.LUN = 000042          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025     F.NREC= 000030          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025     F.PREV= 000000          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025     F.PRNT= 100000          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025     F.PROT= 010000          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025 M.GDWN= 000003      ND.$ES= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025 M.IBM = 000003      ND.$NS= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
F.ERR = 000052          GNOME   003102R 025     MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025     MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
```

CRTFIL MACRO M1113 08-FEB-80 11:30 PAGE 46-3
SYMBOL TABLE

```
NUMQFL  003234R     025 RANDAC  003064R     025 R.LNUM= 000010          SY.BRK= 100000          T.LIST= 000400
NUMRIG  003231R     025 RANDY   003062R     025 R.NEXT= 000006          SY.BS = 000010          T.LIT = 000020
NUMWID  003226R     025 RANERR= 104411          R.OFF = 000000          SY.CLR= 000040          T.MON = 000010
NUMZFL  003235R     025 RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.NIL = 000004
NYIERR= 104432          RARGCV  003220R     025 R.PTR = 000000          SY.NOU= 000100          T.NUM = 002000
N.BLSZ= 000036          RARGFL  003202R     025 R.SEQ = 000003          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLE  003216R     025 R.VAR = 000002          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGLN  003210R     025 R6    =%000006          SY.SRT= 000400          T.SOP = 000002
N.DEVN= 000034          RARGPT  003206R     025 R7    =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGSF  003214R     025 SI    =%000005          SY.SVL= 000200          T.TMP = 000200
N.DVNM= 000032          RARGTY  003212R     025 SIDAMN= 104433          S.AB  = 177776          T.VAL = 100000
N.EXT = 000014          RARGVA  003222R     025 SIFRAM  003314R   025 S.ASB = 100000          T.VAR = 001000
N.FID = 000000          RBASE   005074R     025 SISTAK  006350R   025 S.BFHD= 000020          UPPER = 177400
N.FNAM= 000006          RCOUNR  005112R     025 SITOP   006346R   025 S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          RCOUNT  005110R     025 SITP    007144R   025 S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDCNT  005070R     025 SIWARN= 000200          S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDIGN  005066R     025 SIZTAB  000044R   025 S.FDB = 000140          VARBLS  000000R     025
N.NEXT= 000022          REDOFF  005072R     025 SI.BIT= 177774          S.FLB = 040000          VERNUM  000000R     025
N.PPN = 000024          REENTR  003060R     025 SI.LEN= 177772          S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESBIT  003142R     025 SI.RET= 177772          S.FNB = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESCHN= ****** GX       SI.VAL= 177776          S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESCVT  003152R     025 SKPSIZ  005134R   025 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESERR= 104440          SPHOLD  000004R   024 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESLEN  003150R     025 SPINIT  000002R   024 S.LEN = 000002          WSNAME  000074R     025
OPNSET= ****** GX       RESPTR  003140R     025 STKINC= 000774          S.LINK= 177774          WSTIME  000132R     025
OPTMP0  005066R     025 RESSFT  003146R     025 STOP  = 104420          S.NFEN= 000020          $FIS  = ****** G
OPTMP1  005070R     025 RESTYP  003144R     025 SUBCNT  003104R   025 S.TYPE= 140000          $FPP  = ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBDCT  003106R   025 S.VAL = 000000          $FPPLN= ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBPTR  003110R   025 TKDIMS  005072R   025 $MACHN= ****** G
OPTMP4  005076R     025 RINC    005104R     025 SYMB1   005344R   025 TKFRAM  005066R   025 $RSTS = ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMF    005340R   025 TKLEFT  005070R   025 $RSX  = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMPT1  005346R   025 TMODE   000403R   023 $RT11 = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMTAB  005352R   025 TMPTAG= 000166          .BOOLE= 000002
OPTMP8  005106R     025 ROVER   003312R     025 SYMTBF= 104404          TTYBUF  004140R   025 .CHARA= 000000
OPTMP9  005110R     025 RSTS  = 000000          SYMTOP  005350R   025 TTYLEN= 000622          .DEF. = 000002
OPTM10  005112R     025 RSX   = 000115          SYMTP   006272R   025 TURKEY= 104436          .FLOAT= 000006
OPTM14  005026R     025 RTOP    005070R     025 SYMWRN= 000054          TYPE  = 000166          .FSRCB= ****** G
ORIGIN  000030R     025 RT11  = 000000          SYNERR= 104407          T.BOL = 020000          .INTEG= 000004
OUTPTR  000066R     025 R.BITS= 000003          SYSERR= 104402          T.CON = 004000          .MOLUN  000404R     023
OVROPR= 000000          R.FIX = 000001          SYSTAT  000000R   024 T.DIM = 000100          ...GBL= 000000
PATCH   007344R     025 R.LCT = 000002          SY.BAD= 000001          T.FUN = 010000          ...TPC= 000020
PRECSN  000052R     025 R.LN  = 000002

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
```

```
CRTFIL  MACRO M1113  08-FEB-80 11:30  PAGE 46-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000136      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12948 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:23
CRTFIL,DM1:[305,301]CRTFIL/-SP=APLRX1,APLDEF,CRTFIL,END
```

FILEID**DATA

```
DDDDDDDD        AAAAAA      TTTTTTTTTT      AAAAAA
DDDDDDDD        AAAAAA      TTTTTTTTTT      AAAAAA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AA      AA        TT        AA      AA
DD      DD    AAAAAAAAAA        TT        AAAAAAAAAA
DD      DD    AAAAAAAAAA        TT        AAAAAAAAAA
DD      DD    AA      AA        TT        AA      AA        ....
DD      DD    AA      AA        TT        AA      AA        ....
DDDDDDDD      AA      AA        TT        AA      AA        ....
DDDDDDDD      AA      AA        TT        AA      AA        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:33
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    DATA             DATA AREA FOR SAVE/LOAD/COPY ROUTINES
```

DATA MACRO M1113 08-FEB-80 11:33 PAGE 1

```
      1            000000                  RT11    =  0          ;NOT RT-11
      2            000000                  RSTS    =  0          ;NOT RSTS/E
      3            000115                  RSX     =  'M         ;RSX-11/M
      4
      5            000000                  EIS     =  0          ;NO EIS
      6            000000                  FIS     =  0          ;NO FIS
      7            000000                  FPP     =  0          ;NO FPP
      8
      9            000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11            000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
DATA    MACRO M1113  08-FEB-80 11:33  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  DATA
                                .IDENT  \V02.03\
                                .SBTTL  DATA            DATA AREA FOR SAVE/LOAD/COPY ROUTINES
                                        .PSECT  DATA,   RW, D
                                CPOUTL::
                                        .BLKB   OBUFLN          ;The data area is here
```

```
DATA    MACRO M1113  08-FEB-80 11:33  PAGE 45
DATA             DATA AREA FOR SAVE/LOAD/COPY ROUTINES

      1          000001                              .END
```

DATA MACRO M1113 08-FEB-80 11:33 PAGE 45-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLOK= 000064    D.DECD= 000002    EIS    = 000000
AC1    =%000001          CHRLPR= 000072    C.BLSZ= 000140    D.DIV = 000000    ELESIZ 005132R  025
AC2    =%000002          CHRMNS= 000137    C.BUFF= 177760    D.DMD = 000010    ENDCOR 000004R  025
AC3    =%000003          CHRMUL= 000075    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC4    =%000004          CHRNEG= 000100    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CPTR= 177764    D.ENCD= 000002    ERRNUM 004772R  025
ALLINT 003274R  025      CHRPLU= 000055    C.CZ  = 001000    D.EQ  = 000000    ERRPTR 004770R  025
ARYRET 003276R  025      CHRPWR= 000120    C.DIG = 000040    D.EXPD= 000002    ERR0   004774R  025
ASCQUO= 000042           CHRQST= 000121    C.FASW= 020000    D.FGET= 000005    EXEBUF 005054R  025
A.ALEN= 000004           CHRQUO= 000113    C.FLEN= 000010    D.FMT = 000004    E$CT   005056R  025
A.ARG = 000040           CHRRBR= 000047    C.FLSH= 004000    D.FPNT= 000005    E$PT   005060R  025
A.ASB = 000200           CHRRPR= 000042    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BLZBUB 005116R  025      CHR.LF= 000012    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
BRSELF 005046R  025      CHR.LT= 000043    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
CCNT   005072R  025      CHR.SI= 000017    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CHANEX 003060R  025      CHR.SL= 000057    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHAN00 000256R  025      CHR.SO= 000016    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN01 000330R  025      CHR.TB= 000011    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN02 000512R  025      CHR.UL= 000106    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN03 000674R  025      CHR..A= 000141    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN04 001056R  025      CHR..E= 000145    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN05 001240R  025      CHR..S= 000163    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN06 001422R  025      CHR..T= 000164    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN07 001604R  025      CHR..U= 000165    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN08 001766R  025      CHR..Z= 000172    DEFVIL 005114R  025 D.QUAD= 000000  FD.F11= 040000
CHAN09 002150R  025      CHR..0= 000060    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN10 002332R  025      CHR..1= 000061    DIGITS 000006R  025 D.RHO = 000000  FD.ISP= 002000
CHAN11 002514R  025      CH.AND= 000001    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN12 002676R  025      CMDBUF 003250R  025 DIMBLK 003112R  025 D.SCON= 000006 FD.OSP= 004000
CHAN13 000022R  023      CMDMAX= 000010    DIMPTR 003114R  025 D.SDIF= 000006  FD.PLC= 000004
CHAN14 000204R  023      CNTSIZ 005130R  025 DIMTYP 003116R  025 D.SUB = 000000 FD.PRN= 000004
CHNOPN= 104471           COMBIN 005076R  025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051           COMCNT 005126R  025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133           COMPTR 005122R  025 DYDCNT 003224R  025 D.UNON= 000006 FD.REC= 000001
CHRBOL= 000200           COMRTN 003156R  025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114           COMSKP 005124R  025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054           CORPTR 005120R  025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076           CPOUTL 000000RG 026 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.ASS= 000200      D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.BOO= 000010      D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176           CS.FLO= 000020      D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110           CS.IDN= 000040      D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056           CS.ONE= 000100      D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105           CTEPS  000152R  025 D.COMP= 000001    EDICHR 005063R  025 FF.RWF= 000006
CHRERR= 104431           CTMAX  000212R  025 D.CONT= 000006    EDICNT 005064R  025 FF.SPC= 000004
CHRFIL= 000177           CTRMAX 000172R  025 D.CTLM= 000000    EDIFLG 005062R  025 FHEADR 005012R  025
CHRFRO= 000135           C.ALOC= 000162      D.DEAL= 000003    EDIT  = 000003      FHRET  005040R  025
```

SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464          MONSFT  003126R  025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540          MONTYP  003124R  025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565          MONVAL  003134R  025 M.SIGN= 000000
FIXPTR  003074R  025 F.FUN = 000010          INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R  025 F.FVER= 000120          INNTMP  003250R  025 M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R  025 F.HEAD= 000006          INNTM2  005016R  025 M.BXAV= 000007          M.SYN = 000201
FLINE   005014R  025 F.HIBK= 000004          INPTR   000064R  025 M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R  025 F.LBL = 000020          INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R  025 F.LINE= 000012          IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R  025 F.LOC = 000004          I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R  025 F.LUN = 000042          I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R  025 F.MBCT= 000054          I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R  025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R  025 M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R  025 F.NREC= 000030          LARGBL  003162R  025 M.BXWA= 000007          NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R  025 M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R  025 M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R  025 F.PREV= 000000          LARGLE  003174R  025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R  025 F.PRNT= 100000          LARGLN  003166R  025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R  025 F.PROT= 010000          LARGPT  003164R  025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R  025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R  025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R  025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R  025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R  025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R  025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R  025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R  025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R  025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R  025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R  025 M.INSP= 000006          NEGSGN  005063R  025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R  025 M.IOTA= 000000          NINPT1  003302R  025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020          M.LIT = 000205          NINPT2  003300R  025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R  025 M.LN  = 000001          NINSTK  003304R  025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541          M.MMD = 000010          NINTMP  003310R  025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R  025 M.NEG = 000000          NINTYP  003306R  025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R  025 M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R  025 M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R  025 M.NUM = 000204          NPRT1   004762R  025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012          NULL    007364R  025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R  025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R  025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R  025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R  025
F.ERR = 000052          GNOME   003102R  025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R  025
F.FACC= 000043          HASHC   005136R  025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R  025
F.FFBY= 000014          HASHTB  005140R  025 MONBLK  003120R  025 M.RDCT= 000011          NUMQFL  003234R  025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R  025 M.REVS= 000003          NUMRIG  003231R  025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R  025 M.ROLL= 000000          NUMWID  003226R  025
F.FNAM= 000110          ICHAN   005050R  025 MONPTR  003122R  025 M.SATA= 000001          NUMZFL  003235R  025
```

SYMBOL TABLE

```
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.CLR= 000040          T.LIT = 000020
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR = 000000          SY.ERP= 000020          T.MON = 000010
N.DESC= 000020          RARGLE  003216R     025 R.SEQ = 000003          SY.NOU= 000100          T.NIL = 000004
N.DEV = 000032          RARGLN  003210R     025 R.VAR = 000002          SY.OPN= 000002          T.NUM = 002000
N.DEVN= 000034          RARGPT  003206R     025 R6    =%000006          SY.PMT= 001000          T.OPFN= 000001
N.DID = 000024          RARGSF  003214R     025 R7    =%000007          SY.SRT= 000400          T.OTHR= 040000
N.DVNM= 000032          RARGTY  003212R     025 SI    =%000005          SY.SVC= 000004          T.SOP = 000002
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433          SY.SVL= 000200          T.SPOP= 000040
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R   025 S.AB  = 177776          T.TMP = 000200
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R   025 S.ASB = 100000          T.VAL = 100000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R   025 S.BFHD= 000020          T.VAR = 001000
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R   025 S.BITS= 000002          UPPER = 177400
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200          S.CHR1= 000003          UTNAME  000344R     023
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R   025 S.FATT= 000016          VALERR= 104413
N.PPN = 000024          REENTR  003060R     025 SI.BIT= 177774          S.FDB = 000140          VARBLE  007400R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772          S.FLB = 040000          VARBLS  000000R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.RET= 177772          S.FNAM= 000006          VERNUM  000000R     025
OBUFLN= 001000          RESERR= 104440          SI.VAL= 177776          S.FNB = 000036          VRSION= 000002
OCHAN   005052R   025  RESLEN  003150R     025 SKPSIZ  005134R   025 S.FNBW= 000017          WIDTH   000072R     025
OMODE   000402R   023  RESPTR  003140R     025 SPHOLD  000004R   024 S.FNTY= 000004          WROUND  000020R     025
OPTMP0  005066R   025  RESSFT  003146R     025 SPINIT  000002R   024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP1  005070R   025  RESTYP  003144R     025 STKINC= 000774          S.LEN = 000002          WSFULL= 104403
OPTMP2  005072R   025  RESULT  003136R     025 STOP  = 104420          S.LINK= 177774          WSNAME  000074R     025
OPTMP3  005074R   025  RESVAL  003154R     025 SUBCNT  003104R   025 S.NFEN= 000020          WSTIME  000132R     025
OPTMP4  005076R   025  RINC    005104R     025 SUBDCT  003106R   025 S.TYPE= 140000          $FIS  = ****** G
OPTMP5  005100R   025  RLTBAS  005066R     025 SUBPTR  003110R   025 S.VAL = 000000          $FPP  = ****** G
OPTMP6  005102R   025  ROTINC  005070R     025 SYMB1   005344R   025 TKDIMS  005072R   025 $FPPLN= ****** G
OPTMP7  005104R   025  ROUND   000010R     025 SYMF    005340R   025 TKFRAM  005066R   025 $MACHN= ****** G
OPTMP8  005106R   025  ROVER   003312R     025 SYMPT1  005346R   025 TKLEFT  005070R   025 $RSTS = ****** G
OPTMP9  005110R   025  RSTS  = 000000          SYMTAB  005352R   025 TMODE   000403R   023 $RSX  = ****** G
OPTM10  005112R   025  RSX   = 000115          SYMTBF= 104404          TMPTAG= 000166          $RT11 = ****** G
OPTM14  005026R   025  RTOP    005070R     025 SYMTOP  005350R   025 TTYBUF  004140R   025 .BOOLE= 000002
ORIGIN  000030R   025  RT11  = 000000          SYMTP   006272R   025 TTYLEN= 000622          .CHARA= 000000
OUTPTR  000066R   025  R.BITS= 000003          SYMWRN= 000054          TURKEY= 104436          .DEF. = 000002
OVROPR= 000000          R.FIX = 000001          SYNERR= 104407          TYPE  = 000166          .FLOAT= 000006
PATCH   007344R   025  R.LCT = 000002          SYSERR= 104402          T.BOL = 020000          .FSRCB= ****** G
PRECSN  000052R   025  R.LN  = 000002          SYSTAT  000000R   024 T.CON = 004000          .INTEG= 000004
RANDAC  003064R   025  R.LNUM= 000010          SY.BAD= 000001          T.DIM = 000100          .MOLUN  000404R     023
RANDY   003062R   025  R.NEXT= 000006          SY.BRK= 100000          T.FUN = 010000          ...GBL= 000000
RANERR= 104411          R.OFF = 000000          SY.BS = 000010          T.LIST= 000400          ...TPC= 000020
RARGBL  003204R   025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
DATA    MACRO M1113  08-FEB-80 11:33  PAGE 45-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000000      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
DATA    001000      026
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12896 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:34
DATA,DM1:[305,301]DATA/-SP=APLRX1,APLDEF,DATA,END
```

FILEID**DATIME

```
DDDDDDDD        AAAAAA      TTTTTTTTTT      IIIIII      MM      MM    EEEEEEEEEE
DDDDDDDD        AAAAAA      TTTTTTTTTT      IIIIII      MM      MM    EEEEEEEEEE
DD      DD     AA    AA        TT            II        MMMM  MMMM    EE
DD      DD     AA    AA        TT            II        MMMM  MMMM    EE
DD      DD     AA    AA        TT            II        MM  MM  MM    EE
DD      DD     AA    AA        TT            II        MM  MM  MM    EE
DD      DD     AA    AA        TT            II        MM      MM    EEEEEEEE
DD      DD     AA    AA        TT            II        MM      MM    EEEEEEEE
DD      DD     AAAAAAAAAA      TT            II        MM      MM    EE
DD      DD     AAAAAAAAAA      TT            II        MM      MM    EE
DD      DD     AA    AA        TT            II        MM      MM    EE                ....
DD      DD     AA    AA        TT            II        MM      MM    EE                ....
DDDDDDDD       AA    AA        TT          IIIIII      MM      MM    EEEEEEEEEE        ....
DDDDDDDD       AA    AA        TT          IIIIII      MM      MM    EEEEEEEEEE        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 11:35
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
     2-    29    TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
     2-    33    AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    NEWDT            PRINT CURRENT DATE & TIME
```

DATIME MACRO M1113 08-FEB-80 11:35
TABLE OF CONTENTS

44- 14 DATIME PRINT DATE & TIME

```
DATIME  MACRO M1113  08-FEB-80 11:35  PAGE 1


                                RT11    =  0            ;NOT RT-11
                                RSTS    =  0            ;NOT RSTS/E
                                RSX     = 'M            ;RSX-11/M

                                EIS     =  0            ;NO EIS
                                FIS     =  0            ;NO FIS
                                FPP     =  0            ;NO FPP

                                FPPLEN  =  4.           ;SINGLE PRECISION

                                DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
DATIME  MACRO M1113  08-FEB-80 11:35  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  DATIME
                                        .IDENT  \V02.03\

                                        .SBTTL  NEWDT           PRINT CURRENT DATE & TIME

                                        ;+
                                        ; THIS ROUTINE COMPUTES AND THE PRINTS THE CURRENT DATE & TIME
                                        ;-
000000                                          .PSECT  CODE

000000                                  NEWDT:: CALL    GETMDT          ;COMPUTE CURRENT DATE, TIME


                                        .SBTTL  DATIME          PRINT DATE & TIME

                                        ;+
                                        ; NORMAL ENTRY POINT
                                        ;-
000004                                  DATIME::
000004                                          PUSH    <R0,R1,R2,R3,R4>;SAVE REGISTERS
000016  012700  005066'                         MOV     #OPTMP0,R0      ;POINT TO WORK AREA
000022  012701  000003                         MOV     #3.,R1          ;LOOP COUNT
000026  012702  000132'                         MOV     #WSTIME,R2      ;POINTER TO DATA
000032  112203                          1$:     MOVB    (R2)+,R3        ;GET THE NEXT TIME ITEM
000034                                          CALL    NUMOVE          ;MOVE TWO DIGITS
000040  112720  000076                         MOVB    #CHRCOL,(R0)+   ;':'
000044                                          SOB     R1,1$           ;AND LOOP (3 TIMES)
000050  112760  000040  177777                 MOVB    #CHRSPA,-1(R0)  ;A SPACE
000056  112720  000040                         MOVB    #CHRSPA,(R0)+   ;' '
000062  012701  000141'                         MOV     #WSDATE+3,R1    ;DATE FOR MESSAGE
000066  114103                                 MOVB    -(R1),R3        ;GET THE DAY
000070                                          CALL    NUMOVE          ;PRINT IT
000074  112720  000137                         MOVB    #CHRMNS,(R0)+   ;'-'
000100  114103                                 MOVB    -(R1),R3        ;GET THE MONTH
000102  006303                                 ASL     R3              ;TIMES 2
000104  016302  000000'                         MOV     MONTHT(R3),R2   ;    GET THE MONTH (RAD50)
000110                                          CALL    RAD$            ;    CONVERT
000114  112720  000137                         MOVB    #CHRMNS,(R0)+   ;'-'
000120  114103                                 MOVB    -(R1),R3        ;GET THE YEAR
000122                                          CALL    NUMOVE          ;PRINT THE YEAR
000126  012701  005066'                         MOV     #OPTMP0,R1      ;AGAIN,POINT TO TEMPORARY
000132  122711  000060                         CMPB    #CHR..0,@R1     ;IS IT A LEADING 0 ?
000136  001002                                 BNE     2$              ;NO, IS OK
000140  112711  000040                         MOVB    #CHRSPA,@R1     ;CONVERT TO A SPACE
000144  122761  000060  000012  2$:     CMPB    #CHR..0,12(R1)  ;DATE START WITH A '0'?
000152  001003                                 BNE     3$              ;NO, IS OK
000154  112761  000040  000012                 MOVB    #CHRSPA,12(R1)  ;CONVERT TO SPACE
000162  105061  000023          3$:     CLRB    23(R1)          ;END THE STRING
000166                                          CALL    PRINTM          ;PRINT IT OUT (FINALLY)
000172                                          POP     <R4,R3,R2,R1,R0>;RESTORE REGISTERS
000204                                          RETURN                  ;AND RETURN
```

```
DATIME  MACRO M1113  08-FEB-80 11:35  PAGE 45
DATIME          PRINT DATE & TIME


                                        ;+
                                        ; THIS ROUTINE MOVES A TWO-DIGIT (BASE 10) NUMBER TO
                                        ; THE AREA POINTED TO BY R0.
                                        ;-
000206                          NUMOVE: PUSH    <R2>            ;SAVE HIGH PART OF DBL-WORD
000210  005002                          CLR     R2              ;CLEAR HIGH-WORD
000212                                  DIV     #10.,R2         ;DIVIDE BY TEN - GET TWO NUMBERS
000226  062702  000060                  ADD     #CHR..0,R2      ;CONVERT BOTH TO
000232  062703  000060                  ADD     #CHR..0,R3      ;   ASCII
000236  110220                          MOVB    R2,(R0)+        ;AND MOVE BOTH TO
000240  110320                          MOVB    R3,(R0)+        ;   TEMPORARY
000242                                  POP     R2              ;RESTORE R2
000244                                  RETURN

000000                                  .PSECT  TABLES

000000  134745                  MONTHT: .WORD   35*50+35*50+35  ;MONTH TABLE
000002  037266  023112  050572          .RAD50  /JANFEBMARAPRMAYJUN/
000010  004322  050601  040726
000016  040724  004617  073630          .RAD50  /JULAUGSEPOCTNOVDEC/
000024  057114  054756  014713
```

DATIME PRINT DATE & TIME

```
                                        ;+
                                        ; GET THE CURRENT TIME, DATE
                                        ;-
                                                .PSECT  CODE

                                        GETMDT::
                                                PUSH    R0              ;SAVE REGS
                                                SUB     #16.,SP         ;SPACE FOR PARAMETERS
                                                MOV     SP,R0           ;COPY POINTER TO PARAM SPACE
                                                GTIM$S  R0              ;GET THE PARAMETERS
                                                MOV     R0,-(SP)
                                                MOV     (PC)+,-(SP)
                                                .BYTE   61.,2
                                                EMT     ^O<377>
                                                MOV     #WSDATE,R0      ;POINTER TO DATE FIELD
                                                MOVB    (SP)+,(R0)+     ;GET THE YEAR
                                                MOVB    (SP)+,(R0)+     ;     THE MONTH
                                                MOVB    (SP)+,(R0)+     ;     THE DAY
                                                MOV     #WSTIME,R0      ;POINTER TO TIME FIELD
                                                MOVB    (SP)+,(R0)+     ;GET THE HOUR
                                                MOVB    (SP)+,(R0)+     ;     THE MINUTE
                                                MOVB    (SP)+,(R0)+     ;     THE SECOND
                                                ADD     #4.,SP          ;SKIP TICK INFO
                                                POP     R0              ;RESTORE R0
                                                RETURN
```

```
DATIME  MACRO M1113  08-FEB-80 11:35  PAGE 48
DATIME          PRINT DATE & TIME

      1         000001                          .END
```

DATIME MACRO M1113 08-FEB-80 11:35 PAGE 48-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DECD= 000002    EIS    = 000000
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DIV = 000000    ELESIZ 005132R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DMD = 000010    ENDCOR 000004R     025
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DQAD= 177777    EOF    = 104475
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.ENCD= 000002    ERRNUM 004772R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.EQ  = 000000    ERRPTR 004770R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.EXPD= 000002    ERR0   004774R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FGET= 000005    EXEBUF 005054R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FMT = 000004    E$CT   005056R     025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPNT= 000005    E$PT   005060R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.FPUT= 000005    FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GE  = 000000    FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.GT  = 000000    FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IBM = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.IND = 000002    FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INNR= 000013    FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.INTR= 000006    FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LE  = 000000    FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LOG = 000001    FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.LT  = 000000    FA.RD = 000001
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MAX = 000000    FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MEM = 000002    FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MIN = 000000    FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.MUL = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.NE  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NN  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.NR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.OR  = 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PLUS= 000000    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    DATIME  000004RG  020 D.PSUB= 000006    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVIL   005114R  025 D.QUAD= 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R  025 D.RHO = 000000    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R  025 DIMBLK  003112R  025 D.SCON= 000006    FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010    DIMPTR  003114R  025 D.SDIF= 000006    FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R  025 DIMTYP  003116R  025 D.SUB = 000000    FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R  025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R  025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425    DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R  025 DYDCNT  003224R  025 D.UNON= 000006    FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R  025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R  025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R  025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200    D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040    D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLT= 000110           CS.ONE= 000100    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R  025 D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHREP = 000105           CTMAX   000212R  025 D.COMP= 000001    EDICHR  005063R  025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R  025 D.CONT= 000006    EDICNT  005064R  025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162    D.CTLM= 000000    EDIFLG  005062R  025 FHEADR 005012R     025
CHRFRO= 000135           C.BLOK= 000064    D.DEAL= 000003    EDIT  = 000003       FHRET  005040R     025
```

DATIME MACRO M1113 08-FEB-80 11:35 PAGE 48-2
SYMBOL TABLE

```
FILNTF= 104467          F.FNB  = 000102      G.TIMI= 000010          L.PRNT= 000001          M.PITM= 000000
FIS    = 000000          F.FNDF= 000010      G.TIMO= 000002          L.PTR = 000000          M.PRDS= 000006
FISCON= 046000          F.FTYP= 000116      G.TISC= 000012          L.PTR1= 000004          M.QQAD= 000000
FIXPTR  003074R  025    F.FUN = 000010      G.TIYR= 000000          L.SUBS= 000002          M.QUAD= 000000
FLAGS   004764R  025    F.FVER= 000120      HASHC   005136R 025    MACHIN= 000106          M.RAND= 000012
FLCBUF  003250R  025    F.HEAD= 000006      HASHTB  005140R 025    MAXCHN= 000014          M.RAVL= 000000
FLINE   005014R  025    F.HIBK= 000004      HSHMSK= 177601          MONBLK  003120R 025    M.RCIP= 000000
FLOEXP  003270R  025    F.LBL = 000020      HSHSIZ= 000200          MONCVT  003132R 025    M.RDCT= 000011
FLONEG  003272R  025    F.LINE= 000012      ICHAN   005050R 025    MONLEN  003130R 025    M.REVS= 000003
FLSWTB  000222R  025    F.LOC = 000004      IDIV  = ****** GX       MONPTR  003122R 025    M.ROLL= 000000
FLTLEN= 000010          F.LSTC= 000004      ILFNAM= 104464          MONSFT  003126R 025    M.SATA= 000001
FNINSI  005044R  025    F.LUN = 000042      ILLCHN= 104540          MONTHT  000000R 002    M.SCAN= 000011
FORIGN  000032R  025    F.MBCT= 000054      ILLSW = 104565          MONTYP  003124R 025    M.SHAP= 000000
FO.APD= 000106          F.MBC1= 000055      INDERR= 104410          MONVAL  003134R 025    M.SIGN= 000000
FO.MFY= 000002          F.MBFG= 000056      INNTMP  003250R 025    M.ABS = 000000          M.SUBR= 000001
FO.RD = 000001          F.MON = 000002      INNTM2  005016R 025    M.BOL = 000200          M.SYM = 000203
FO.UPD= 000006          F.NIL = 000001      INPTR   000064R 025    M.BXAV= 000007          M.SYN = 000201
FO.WRT= 000016          F.NRBD= 000024      INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FPMODE  000042R  025    F.NREC= 000030      IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FPP    = 000000          F.OPN = 000010      I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FPPLEN= 000004          F.OVBS= 000030      I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FREBLK  007400R  025    F.PREV= 000000      I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FUZZ    000142R  025    F.PRNT= 100000      I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FXPTR1  003076R  025    F.PROT= 010000      I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
F$EVAL= 000100          F.RACC= 000016      I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
F$EVI  = 000200          F.RATT= 000001      I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
F$EXEC= 000400          F.RCNM= 000034      LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
F$FUNC= 001000          F.RCTL= 000017      LABLPT  005042R 025    M.BXRL= 000007          NAMBUF  003240R 025
F.ACTL= 000076          F.READ= 004000      LARGBL  003162R 025    M.BXTS= 000007          NB.DEV= 000200
F.ALOC= 000040          F.REDF= 000040      LARGCV  003176R 025    M.BXWA= 000007          NB.DIR= 000100
F.BBFS= 000062          F.RES  = 000040      LARGFL  003160R 025    M.CEI  = 177777          NB.NAM= 000004
F.BDB  = 000070          F.RET  = 177772      LARGLE  003174R 025    M.CHR  = 000202          NB.SD1= 000400
F.BGBC= 000057          F.RSIZ= 000002      LARGLN  003166R 025    M.CMT  = 000206          NB.SD2= 001000
F.BITS= 177774          F.RTYP= 000000      LARGPT  003164R 025    M.DOUT= 000006          NB.SNM= 000040
F.BKDN= 000026          F.R2   = 000014      LARGSF  003172R 025    M.DQAD= 000006          NB.STP= 000020
F.BKDS= 000020          F.R3   = 000016      LARGTY  003170R 025    M.ENCD= 000004          NB.SVR= 000010
F.BKEF= 000050          F.R4   = 000020      LARGVA  003200R 025    M.EXE  = 000000          NB.TYP= 000002
F.BKP1= 000051          F.SEQN= 000100      LBINC   005100R 025    M.EXP  = 000001          NB.VER= 000001
F.BKST= 000024          F.SPDV= 000072      LBUFF   003316R 025    M.F CT= 000001          ND.DEV= 000200
F.BKVB= 000064          F.SPUN= 000074      LBUFLN= 000621          M.FIL  = 000177          ND.EXT= 000002
F.BR   = 000020          F.STBK= 000036      LCOUNT  005106R 025    M.FLR  = 177777          ND.NAM= 000004
F.CHR  = 000075          F.STK  = 000022      LENERR= 104412          M.FMT  = 000004          ND.PPN= 000100
F.CNTG= 000034          F.STOP= 000004      LENTAB  000054R 025    M.GDUP= 000003          ND.$EX= 000000
F.CODE= 000010          F.STP  = 000001      LEXPTR  003100R 025    M.GDWN= 000003          ND.$ES= 000020
F.DFNB= 000046          F.SUSP= 000002      LFTOP   005066R 025    M.GETI= 000006          ND.$NM= 000000
F.DSPT= 000044          F.TRC  = 000002      LINC    005102R 025    M.IBM  = 000003          ND.$NS= 000040
F.DVNM= 000134          F.UNIT= 000136      LINE    005016R 025    M.IDN  = 000000          ND.$PN= 001400
F.EFBK= 000010          F.URBD= 000020      LININC  005026R 025    M.INSP= 000006          NEGSGN  005063R 025
F.EFN  = 000050          F.VAL  = 177776      LISIZE= 000020          M.IOTA= 000000          NEWDT   000000RG 020
F.END  = 002000          F.VBN  = 000064      LITOP   005072R 025    M.LIT  = 000205          NINPT1  003302R 025
F.EOBB= 000032          F.VBSZ= 000060      LNLNG  = 104541          M.LN   = 000001          NINPT2  003300R 025
F.ERR  = 000052          GETMDT  000246RG 020 LOCLCT  005036R 025    M.MMD  = 000010          NINSTK  003304R 025
F.FACC= 000043          GNOME   003102R 025 LOWCOR  000002R 025    M.NEG  = 000000          NINTMP  003310R 025
F.FFBY= 000014          G.TICP= 000016      LSTERP  004766R 025    M.NOT  = 000000          NINTYP  003306R 025
F.FLAG= 000002          G.TICT= 000014      LWIDTH  000070R 025    M.NUB  = 000006          NOTAVL= 104472
F.FLB  = 000100          G.TIDA= 000004      L.BITS= 000003          M.NUM  = 000204          NOTOPN= 104473
F.FNAM= 000110          G.TIHR= 000006      L.CNT  = 000002          M.OUTR= 000012          NPRT1   004762R 025
```

SYMBOL TABLE

```
NULL    007364R  025 OPTM14  005026R  025 RTOP    005070R  025 SYSERR= 104402          T.FUN = 010000
NUMCHN= 000020       ORIGIN  000030R  025 RT11  = 000000       SYSTAT  000000R  024 T.LIST= 000400
NUMDFL  003230R  025 OUTPTR 000066R  025 R.BITS= 000003       SY.BAD= 000001       T.LIT = 000020
NUMEFL  003233R  025 OVROPR= 000000       R.FIX = 000001       SY.BRK= 100000       T.MON = 000010
NUMESN  003236R  025 PATCH   007344R  025 R.LCT = 000002       SY.BS = 000010       T.NIL = 000004
NUMFLG  003230R  025 PRECSN  000052R  025 R.LN  = 000002       SY.CLR= 000040       T.NUM = 002000
NUMLEN  000062R  025 PRINTM= ****** GX     R.LNUM= 000010       SY.ERP= 000020       T.OPFN= 000001
NUMNEG  003232R  025 RAD$  = ****** GX     R.NEXT= 000006       SY.NOU= 000100       T.OTHR= 040000
NUMOVE  000206R  020 RANDAC  003064R  025 R.OFF = 000000       SY.OPN= 000002       T.SOP = 000002
NUMQFL  003234R  025 RANDY   003062R  025 R.PREV= 000004       SY.PMT= 001000       T.SPOP= 000040
NUMRIG  003231R  025 RANERR= 104411       R.PTR = 000000       SY.SRT= 000400       T.TMP = 000200
NUMWID  003226R  025 RARGBL  003204R  025 R.SEQ = 000003       SY.SVC= 000004       T.VAL = 100000
NUMZFL  003235R  025 RARGCV  003220R  025 R.VAR = 000002       SY.SVL= 000200       T.VAR = 001000
NYIERR= 104432       RARGFL  003202R  025 R6    =%000006       S.AB  = 177776       UPPER = 177400
N.BLSZ= 000036       RARGLE  003216R  025 R7    =%000007       S.ASB = 100000       UTNAME  000344R  023
N.DESC= 000020       RARGLN  003210R  025 SI    =%000005       S.BFHD= 000020       VALERR= 104413
N.DEV = 000032       RARGPT  003206R  025 SIDAMN= 104433       S.BITS= 000002       VARBLE  007400R  025
N.DEVN= 000034       RARGSF  003214R  025 SIFRAM  003314R  025 S.CHR1= 000003       VARBLS  000000R  025
N.DID = 000024       RARGTY  003212R  025 SISTAK  006350R  025 S.FATT= 000016       VERNUM  000000R  025
N.DVNM= 000032       RARGVA  003222R  025 SITOP   006346R  025 S.FDB = 000140       VRSION= 000002
N.EXT = 000014       RBASE   005074R  025 SITP    007144R  025 S.FLB = 040000       WIDTH   000072R  025
N.FID = 000000       RCOUNR  005112R  025 SIWARN= 000200       S.FNAM= 000006       WROUND  000020R  025
N.FNAM= 000006       RCOUNT  005110R  025 SIZTAB  000044R  025 S.FNB = 000036       WSDATE  000136R  025
N.FTYP= 000014       REDCNT  005070R  025 SI.BIT= 177774       S.FNBW= 000017       WSFULL= 104403
N.FVER= 000016       REDIGN  005066R  025 SI.LEN= 177772       S.FNTY= 000004       WSNAME  000074R  025
N.NAME= 000006       REDOFF  005072R  025 SI.RET= 177772       S.FTYP= 000002       WSTIME  000132R  025
N.NEXT= 000022       REENTR  003060R  025 SI.VAL= 177776       S.LEN = 000002       $FIS  = ****** G
N.PPN = 000024       REGTYP= 000002       SKPSIZ  005134R  025 S.LINK= 177774       $FPP  = ****** G
N.STAT= 000020       RESBIT  003142R  025 SPHOLD  000004R  024 S.NFEN= 000020       $FPPLN= ****** G
N.UNIT= 000034       RESCVT  003152R  025 SPINIT  000002R  024 S.TYPE= 140000       $MACHN= ****** G
OBUFLN= 001000       RESERR= 104440       STKINC= 000774       S.VAL = 000000       $RSTS = ****** G
OCHAN   005052R  025 RESLEN  003150R  025 STOP  = 104420       TKDIMS  005072R  025 $RSX  = ****** G
OMODE   000402R  023 RESPTR  003140R  025 SUBCNT  003104R  025 TKFRAM  005066R  025 $RT11 = ****** G
OPTMPC  005066R  025 RESSFT  003146R  025 SUBDCT  003106R  025 TKLEFT  005070R  025 $$$OST= 000020
OPTMP1  005070R  025 RESTYP  003144R  025 SUBPTR  003110R  025 TMODE   000403R  023 .BOOLE= 000002
OPTMP2  005072R  025 RESULT  003136R  025 SYMB1   005344R  025 TMPTAG= 000166       .CHARA= 000000
OPTMP3  005074R  025 RESVAL  003154R  025 SYMF    005340R  025 TTYBUF  004140R  025 .DEF. = 000002
OPTMP4  005076R  025 RINC    005104R  025 SYMPT1  005346R  025 TTYLEN= 000622       .FLOAT= 000006
OPTMP5  005100R  025 RLTBAS  005066R  025 SYMTAB  005352R  025 TURKEY= 104436       .FSRCB= ****** G
OPTMP6  005102R  025 ROTINC  005070R  025 SYMTBF= 104404       TYPE  = 000166       .INTEG= 000004
OPTMP7  005104R  025 ROUND   000010R  025 SYMTOP  005350R  025 T.BOL = 020000       .MOLUN  000404R  023
OPTMP8  005106R  025 ROVER   003312R  025 SYMTP   006272R  025 T.CON = 004000       ...GBL= 000000
OPTMP9  005110R  025 RSTS  = 000000       SYMWRN= 000054       T.DIM = 000100       ...TPC= 000020
OPTM10  005112R  025 RSX   = 000115       SYNERR= 104407

. ABS.  000000  000
        000000  001
TABLES  000032  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
```

```
DATIME  MACRO M1113  08-FEB-80 11:35  PAGE 48-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000322      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13155 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:30
DATIME,DM1:[305,301]DATIME/-SP=APLRX1,APLDEF,DATIME,END
```

```
**FILE**ID**DECNAM
```

```
DDDDDDDD     EEEEEEEEEE     CCCCCCCC   NN      NN     AAAAAA     MM      MM
DDDDDDDD     EEEEEEEEEE     CCCCCCCC   NN      NN     AAAAAA     MM      MM
DD      DD   EE             CC         NN      NN   AA      AA   MMMM  MMMM
DD      DD   EE             CC         NN      NN   AA      AA   MMMM  MMMM
DD      DD   EE             CC         NNNN    NN   AA      AA   MM MM MM MM
DD      DD   EE             CC         NNNN    NN   AA      AA   MM MM MM MM
DD      DD   EEEEEEEE       CC         NN NN   NN   AA      AA   MM    MM MM
DD      DD   EEEEEEEE       CC         NN NN   NN   AA      AA   MM    MM MM
DD      DD   EE             CC         NN   NNNN   AAAAAAAAAA   MM    MM MM
DD      DD   EE             CC         NN   NNNN   AAAAAAAAAA   MM    MM MM
DD      DD   EE             CC         NN     NN   AA      AA   MM    MM MM      ....
DD      DD   EE             CC         NN     NN   AA      AA   MM    MM MM      ....
DDDDDDDD     EEEEEEEEEE     CCCCCCCC   NN     NN   AA      AA   MM    MM MM      ....
DDDDDDDD     EEEEEEEEEE     CCCCCCCC   NN     NN   AA      AA   MM    MM MM      ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:38
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    DECNAM           DECODE FILE NAME
```

```
DECNAM  MACRO M1113  08-FEB-80 11:38  PAGE 1


        1   000000                  RT11    =  0            ;NOT RT-11
        2   000000                  RSTS    =  0            ;NOT RSTS/E
        3   000115                  RSX     = 'M            ;RSX-11/M
        4
        5   000000                  EIS     =  0            ;NO EIS
        6   000000                  FIS     =  0            ;NO FIS
        7   000000                  FPP     =  0            ;NO FPP
        8
        9   000004                  FPPLEN  =  4.           ;SINGLE PRECISION
       10
       11   000000                  DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                        .TITLE  DECNAM
                        .IDENT  \V02.03\

                        .SBTTL  DECNAM          DECODE FILE NAME

                        ;'DECNAM' TAKES AN APL NAME AND CONVERTS IT TO AN INTERNAL FORMAT.
                        ;THE DESCTRIPTION OF THE RETURNED FORMAT CAN BE FOUND IN THE APL COMMON
                        ;DEFINITION FILE UNDER 'NAME BLOCKS'.  THE GENERAL NAME FORMAT IS:
                        ;
                        ;       DEVn:[prj,prg]NAME.EXT {;generation}
                        ;
                        ;THIS FORMAT GETS PRETTY COMPLICATED, BUT NOT ALL OF THESE FIELDS MUST
                        ;BE SPECIFIED. THE MOST COMMONLY USED PART OF THE NAME IS:
                        ;
                        ;       DEVn:NAME.EXT
                        ;
                        ;THIS ROUTINE IS USED ON THE RSX-11/IAS/RSTS/RT-11 VERSIONS OF APL. THE
                        ;PARTS OF THE NAME DESCRIBED BRIEFLY ARE:
                        ;
                        ;DEVn:                  DEVICE NAME AND NUMBER
                        ;
                        ;       THE DEVICE MUST BE TWO LETTERS FOLLOWED BY AN OPTIONAL NUMBER,
                        ;       THEN A SEMI-COLON (:).  IF THE NUMBER IS LEFT OUT, A ZERO IS
                        ;       ASSUMED. IF A NUMBER IS SPECIFIED, IT MUST BE FROM 0 TO 99 FOR RSX.
                        ;
                        ;NAME.EXT               NAME AND EXTENSION
                        ;
                        ;       ALTHOUGH THESE ARE TWO SEPARATE FIELDS, THEY ARE VERY SIMILAR
                        ;       AND ARE SOMETIMES THOUGHT OF AS THE NAME WHEN COMBINED.  THE
                        ;       NAME CAN BE UP TO NINE (SIX FOR RSTS AND RT11) CHARACTERS ALTHOUGH
                        ;       ANYMORE THAN THAT ARE IGNORED.  THE EXTENSION CAN BE UP TO THREE
                        ;       CHARACTERS AND ANY MORE THAN THAT ARE IGNORED.  A SPECIAL CASE OF
                        ;       THE NAME AND EXTENSION IS SIMPLY AN ASTERISK (*), WHICH IS A WILD CARD
                        ;       CHARACTER.  LEGAL CHARACTERS FOR THE NAME AND EXTENSION
                        ;       INCLUDE:    ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
                        ;       THE NAME AND EXTENSION NEED NOT APPEAR TOGETHER AS IN THE EXAMPLE
                        ;       ABOVE, BUT THE DOT (.) MUST ALWAYS PRECEED THE EXTENSION.
                        ;
                        ;[prj,prg]              PROJECT/PROGRAMMER NUMBER
                        ;
                        ;       THIS SPECIFIES THE ACCOUNT ON RSTS.  THE TWO NUMBERS ARE TAKEN
                        ;       MODULO 256 AND ARE PUT TOGETHER IN ONE WORD TO SPECIFY THE
                        ;       ACCOUNT.  SPECIAL CHARACTERS CAN ALSO BE USED TO SPECIFY THE
                        ;       ACCOUNT AND NEED NOT BE ENCLOSED IN SQUARE BRACKETS.  THESE ARE:
                        ;               CHARACTER       ACCOUNT
                        ;               @               ASSIGNED ACCOUNT
                        ;               "               GROUP ACCOUNT [YOUR OWN PRJ#,0]
                        ;               $               [1,2]
                        ;               !               [1,3]
                        ;               %               [1,4]
                        ;               &               [1,5]
                        ;       THE PPN IS IGNORED ON RT11
                        ;
```

```
DECNAM  MACRO M1113  08-FEB-80 11:38  PAGE 45
DECNAM           DECODE FILE NAME

     55 000000                                      .PSECT  CODE
     56
     57 000000                              DECNAM::
     58 000000                                      PUSH    <R0,R1,R2>      ;SAVE LENGTH, POINTER, ETC...
     59 000006  005000                              CLR     R0              ;COUNT OF # OF CHARACTERS
     60 000010                                      CALL    GETNB           ;GET A NON-BLANK CHAR
     61 000014                                      PUSH    <R2,R4>         ;Save Character Data
     62 000020  120227  000015          1$:         CMPB    R2,#CHR.CR      ;A <CR> ?
     63 000024  001413                              BEQ     3$              ;YES, FINISHED
     64 000026  005200                              INC     R0              ;INCREMENT R0
     65 000030  006302                              ASL     R2              ;MOD THE CHARACTER
     66 000032  105762  000001G                     TSTB    ASCTAB+1(R2)    ;TWO-CHAR MNEUMONIC?
     67 000036  001402                              BEQ     2$              ;NOPE
     68 000040  062700  000002                      ADD     #2,R0           ;YES, INCREMENT CHAR
     69 000044                              2$:     GETC    3$              ;GET ANOTHER CHARACTER
        000044  005303                              DEC R3                          ;ANY CHARACTERS LEFT?
        000046  002402                              BLT 3$                  ;NO, BRANCH TO 3$
        000050  112402                              MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
     70 000052  000762                              BR      1$              ;LOOP UNTIL DONE
     71 000054  010001                      3$:     MOV     R0,R1           ;COPY SIZE
     75 000056                                      CALL    GETCOR          ;GET MEMORY, PLEASE
     76 000062                                      SIFRAM  ,#T.TMP         ;Put block on SI
     77 000074                                      POP     <R4,R2>         ;Restore Data Pointers
     78 000100                                      PUSH    R0              ;SAVE STRING POINTER
     79 000102  010103                              MOV     R1,R3           ;Get count to use
     80 000104  005303                              DEC     R3              ;Adjust count, We've used a char already
     81 000106  002446                              BLT     11$             ;Nothing Left, Null string
     83 000110  005046                              CLR     -(SP)           ;USE STACK AS A FLAG
     85 000112  120227  000054          5$:         CMPB    R2,#CHRCMA      ;A COMMA ?
     86 000116  001005                              BNE     6$              ;NO
     88 000120  005716                              TST     @SP             ;WITHIN BRACKETS ?
     89 000122  001014                              BNE     8$              ;YES
     91 000124  012702  000056                      MOV     #CHRDOT,R2      ;NO, TRANSLATE TO ''.''
     92 000130  000411                              BR      8$
     93 000132                              6$:
     94 000132  120227  000073                      CMPB    R2,#CHRLBR      ;LEFT BRACKET ?
     95 000136  001002                              BNE     7$              ;NO
     96 000140  010516                              MOV     SI,@SP          ;FLAG (WITHIN BRACKETS)
     97 000142  000404                              BR      8$
     98 000144  120227  000047          7$:         CMPB    R2,#CHRRBR      ;RIGHT BRACKET ?
     99 000150  001001                              BNE     8$              ;NO
    100 000152  005016                              CLR     @SP
    102 000154  006302                      8$:     ASL     R2              ;MULTIPLY BY 2 FOR WORD OFFSET
    103 000156  016202  000000G                     MOV     ASCTAB(R2),R2   ;GET ASCII MNEUMONIC
    104 000162  032702  177400                      BIT     #UPPER,R2       ;2 LETTERS ?
    105 000166  001406                              BEQ     9$              ;NO...
    106 000170  162703  000002                      SUB     #2,R3           ;COUNT THE EXTRA CHARACTERS
    107 000174  112720  000056                      MOVB    #CHRDOT,(R0)+   ;GET A DOT
    108 000200  110220                              MOVB    R2,(R0)+        ;OUTPUT A CHARACTER
    109 000202  000302                              SWAB    R2              ;SWITCH BACK
    110 000204  110220                      9$:     MOVB    R2,(R0)+        ;STORE THE CHARACTER
    111 000206                                      GETC    10$             ;Get Next Character
        000206  005303                              DEC R3                          ;ANY CHARACTERS LEFT?
        000210  002404                              BLT 10$                 ;NO, BRANCH TO 10$
        000212  112402                              MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
    112 000214  042702  177400                      BIC     #UPPER,R2       ;Clear Upper Byte
    113 000220  000734                              BR      5$              ;LOOP UNTIL DONE
```

```
DECNAM  MACRO M1113  08-FEB-80 11:38  PAGE 45-1
DECNAM           DECODE FILE NAME

    114
    115 000222                                  10$:
    117 000222  005726                                 TST     (SP)+           ;GET RID OF FLAG
    119 000224                                  11$:
    121 000224  012700  000054                       MOV     #C.SIZE,R0      ;GET SIZE OF CSI BLOCK + DUMMY AREA
    122 000230                                       CALL    GETCOR          ;GET IT, PLEASE
    123 000234                                       SIFRAM  ,#T.TMP         ;Build SI Frame Info
    124 000246                                       CSI$1   R0,(SP)+,R1     ;START PARAMETERS
        000246  012660  000004                       MOV     (SP)+,C.CMLD+2(R0)
        000252  010160  000002                       MOV     R1,C.CMLD(R0)
        000256  004737  000000G                      JSR     PC,.CSI1
    125 000262                                       CSI$2   R0,OUTPUT,#FLSWTB       ;GET THE NAME
        000262  112710  000002                     MOVB    #CS.OUT,(R0)
        000266  012760  000222' 000022              MOV     #FLSWTB,C.SWAD(R0)
        000274  004737  000000G                      JSR     PC,.CSI2
    126 000300  103001                               BCC     16$                     ;All ok
    127 000302  104565                               ILLSW                           ;Probably an illegal switch on file spec
    128 000304  016603  000004          16$:         MOV     4(SP),R3                ;Get pointer to FDB
    129 000310  042763  020000  177756               BIC     #C.FASW,C.SBIT(R3)      ;No switch yet
    130 000316  056063  000026  177756               BIS     C.MKW2(R0),C.SBIT(R3)   ;CR/LF's bit
    131 000324                                       POP     R3                      ;POP DATA POINTER
    132 000326  011601                               MOV     (SP),R1                 ;Get Pointer to Name Block
    133 000330  016600  000002                       MOV     2(SP),R0                ;Get Pointer to FDB
    134 000334  016502  177776                       MOV     SI.VAL(SI),R2           ;POINT TO CSI BLOCK
    135 000340  062702  000006                       ADD     #C.DSDS,R2              ;POINT TO GOOD STUFF
    136 000344                                       CALL    .PARSE                  ;PUT STUFF IN PLACE
    137 000350                                       CALL    RSXERR                  ;CHECK FOR ERRORS
    138 000354                                       CALL    KILL                    ;KILL CSI BLOCK
    199 000360                                       CALL    KILL                    ;KILL TRANSLATED STUFF
    200 000364                                       POP     <R1,R0>                 ;Pointer to Name Block, FDB
    201 000370  010002                               MOV     R0,R2                   ;Restore Channel Block Pointer
    202 000372                                       RETURN
```

```
DECNAM  MACRO M1113  08-FEB-80 11:38  PAGE 46
DECNAM           DECODE FILE NAME

      1          000001                          .END
```

DECNAM MACRO M1113 08-FEB-80 11:38 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    CS.NMF= 000001      DIMBLK  003112R 025 D.SCON= 000006
AC1    =%000001          CHRLBR= 000073    CS.ONE= 000100      DIMPTR  003114R 025 D.SDIF= 000006
AC2    =%000002          CHRLPR= 000072    CS.OUT= 000002      DIMTYP  003116R 025 D.SUB = 000000
AC3    =%000003          CHRMNS= 000137    CS.WLD= 000010      DIRFUL= 104466       D.SUBS= 000006
AC4    =%000004          CHRMUL= 000075    CTEPS   000152R 025 DOMERR= 104417       D.TAKE= 000002
AC5    =%000005          CHRNEG= 000100    CTMAX   000212R 025 DPTERR= 104416       D.TRAN= 000004
ALLINT  003274R     025  CHRPDL= 000271    CTRMAX  000172R 025 DYDCNT  003224R 025 D.UNON= 000006
ARYRET  003276R     025  CHRPLU= 000055    C.ALOC= 000162      D.AND = 000000       D.$MAX= 000000
ASCQUO= 000042           CHRPWR= 000120    C.BLOK= 000064      D.BXCT= 000007       D.$MIN= 000000
ASCTAB= ****** GX        CHRQST= 000121    C.BLSZ= 000140      D.BXIO= 000007       D.$MUL= 000000
A.ALEN= 000004           CHRQUO= 000113    C.BUFF= 177760      D.BXNL= 000007       D.$PLU= 000000
A.ARG = 000040           CHRRBR= 000047    C.BUFL= 177762      D.BXPP= 000007       D.$RES= 000000
A.ASB = 000200           CHRRPR= 000042    C.CCNT= 177766      D.BXPW= 000007       D.$SUB= 000000
A.BITS= 000003           CHRSEM= 000074    C.CMLD= 000002      D.BXRL= 000007       D.1COM= 000001
A.CON = 000020           CHRSMO= 000112    C.CPTR= 177764      D.CIRC= 000001       D.1EXP= 000002
A.DIMS= 000002           CHRSPA= 000040    C.CZ  = 001000      D.COMB= 000001       D.1ROT= 000003
A.DIM1= 000006           CHRUPA= 000131    C.DEVD= 000006      D.COMP= 000001       EDICHR  005063R     025
A.NUM = 000006           CHR.BS= 000010    C.DIG = 000040      D.CONT= 000006       EDICNT  005064R     025
A.PTR = 000000           CHR.CC= 000003    C.DIRD= 000012      D.CTLM= 000000       EDIFLG  005062R     025
A.TMP = 000010           CHR.CR= 000015    C.DSDS= 000006      D.DEAL= 000003       EDIT  = 000003
BADDIR= 104463           CHR.CZ= 000032    C.FASW= 020000      D.DECD= 000002       EIS   = 000000
BADERR= 104437           CHR.GT= 000046    C.FILD= 000016      D.DIV = 000000       ELESIZ  005132R     025
BLZBUB  005116R     025  CHR.LF= 000012    C.FLEN= 000010      D.DMD = 000010       ENDCOR  000004R     025
BRSELF  005046R     025  CHR.LT= 000043    C.FLSH= 004000      D.DQAD= 177777       EOF   = 104475
CCNT    005072R     025  CHR.SI= 000017    C.FRC = 000001      D.DROP= 000002       ERRMAX= 000166
CHANEX  003060R     025  CHR.SL= 000057    C.IDEN= 000010      D.ENCD= 000002       ERRNUM  004772R     025
CHAN00  000256R     025  CHR.SO= 000016    C.MKW1= 000024      D.EQ  = 000000       ERRPTR  004770R     025
CHAN01  000330R     025  CHR.TB= 000011    C.MKW2= 000026      D.EXPD= 000002       ERR0    004774R     025
CHAN02  000512R     025  CHR.UL= 000106    C.NAME= 000102      D.FGET= 000005       EXEBUF  005054R     025
CHAN03  000674R     025  CHR..A= 000141    C.NOBF= 000400      D.FMT = 000004       E$CT    005056R     025
CHAN04  001056R     025  CHR..E= 000145    C.NUM = 000020      D.FPNT= 000005       E$PT    005060R     025
CHAN05  001240R     025  CHR..S= 000163    C.OP  = 000004      D.FPUT= 000005       FA.APD= 000100
CHAN06  001422R     025  CHR..T= 000164    C.OPN = 010000      D.GE  = 000000       FA.CRE= 000010
CHAN07  001604R     025  CHR..U= 000165    C.RDLN= 177776      D.GT  = 000000       FA.DLK= 001000
CHAN08  001766R     025  CHR..Z= 000172    C.REWT= 002000      D.IBM = 000002       FA.ENB= 100000
CHAN09  002150R     025  CHR..0= 000060    C.RND = 000010      D.IND = 000002       FA.EXC= 002000
CHAN10  002332R     025  CHR..1= 000061    C.SBIT= 177756      D.INNR= 000013       FA.EXT= 000004
CHAN11  002514R     025  CH.AND= 000001    C.SIZE= 000054      D.INTR= 000006       FA.NSP= 000100
CHAN12  002676R     025  CMDBUF  003250R 025 C.SPFX= 000100    D.LE  = 000000       FA.NSR= 000200
CHAN13  000022R     023  CMDMAX= 000010      C.SPGL= 000200    D.LOG = 000001       FA.POS= 010000
CHAN14  000204R     023  CNTSIZ  005130R 025 C.SPLX= 000001    D.LT  = 000000       FA.RD = 000001
CHNOPN= 104471           COMBIN  005076R 025 C.SPRS= 000002    D.MAX = 000000       FA.RWD= 004000
CHRAND= 000051           COMCNT  005126R 025 C.SRC = 177774    D.MEM = 000002       FA.SEQ= 040000
CHRBEL= 000007           COMERR= 104425      C.SRCP= 000024    D.MIN = 000000       FA.SHR= 000040
CHRBKR= 000133           COMPTR  005122R 025 C.STAT= 000001    D.MUL = 000000       FA.TMP= 000020
CHRBOL= 000200           COMRTN  003156R 025 C.SWAD= 000022    D.NE  = 000000       FA.WCK= 020000
CHRBOX= 000114           COMSKP  005124R 025 C.TT  = 000004    D.NN  = 000000       FA.WRT= 000002
CHRCMA= 000054           CORPTR  005120R 025 C.TYPR= 000000    D.NR  = 000000       FD.BLK= 000010
CHRCOL= 000076           CS.ASS= 000200      C.WIDC= 177770    D.OR  = 000000       FD.CCL= 000002
CHRDEL= 000107           CS.BOO= 000010      C.WIDR= 177772    D.PLUS= 000000       FD.COM= 020000
CHRDIV= 000053           CS.DIF= 000002      DEBUG = 000000    D.PSUB= 000006       FD.CR = 000002
CHRDLR= 000176           CS.DVF= 000004      DECNAM  000000RG 020 D.PWR = 000001    FD.DIR= 000010
CHRDLT= 000110           CS.EQU= 000040      DEFERR= 104405      D.QQAD= 000000       FD.FTN= 000001
CHRDOT= 000056           CS.FLO= 000020      DEVIL   005114R 025 D.QUAD= 000000       FD.F11= 040000
CHREP = 000105           CS.IDN= 000040      DEVNTF= 104470      D.RES = 000000       FD.INS= 000010
CHRERR= 104431           CS.INP= 000001      DIGITS  000006R 025 D.RHO = 000000       FD.ISP= 002000
CHRFIL= 000177           CS.MOR= 000020      DIMAX = 000020      D.ROT = 000003       FD.MNT= 100000
```

SYMBOL TABLE

```
FD.OSP= 004000          F.BKP1= 000051          F.SEQN= 000100          LFTOP   005066R 025 M.IBM = 000003
FD.PLC= 000004          F.BKST= 000024          F.SPDV= 000072          LINC    005102R 025 M.IDN = 000000
FD.PRN= 000004          F.BKVB= 000064          F.SPUN= 000074          LINE    005016R 025 M.INSP= 000006
FD.PSE= 010000          F.BR  = 000020          F.STBK= 000036          LININC  005026R 025 M.IOTA= 000000
FD.RAH= 000001          F.CHR = 000075          F.STK = 000022          LISIZE= 000020      M.LIT = 000205
FD.RAN= 000002          F.CNTG= 000034          F.STOP= 000004          LITOP   005072R 025 M.LN  = 000001
FD.REC= 000001          F.CODE= 000010          F.STP = 000001          LNLNG = 104541      M.MMD = 000010
FD.RWM= 000001          F.DFNB= 000046          F.SUSP= 000002          LOCLCT  005036R 025 M.NEG = 000000
FD.SDI= 000020          F.DSPT= 000044          F.TRC = 000002          LOWCOR  000002R 025 M.NOT = 000000
FD.SQD= 000040          F.DVNM= 000134          F.UNIT= 000136          LSTERP  004766R 025 M.NUB = 000006
FD.TTY= 000004          F.EFBK= 000010          F.URBD= 000020          LWIDTH  000070R 025 M.NUM = 000204
FD.WBH= 000002          F.EFN = 000050          F.VAL = 177776          L.BITS= 000003      M.OUTR= 000012
FF.CHR= 000005          F.END = 002000          F.VBN = 000064          L.CNT = 000002      M.PITM= 000000
FF.NV = 000003          F.EOBB= 000032          F.VBSZ= 000060          L.PRNT= 000001      M.PRDS= 000006
FF.POE= 000002          F.ERR = 000052          GETCOR= ****** GX       L.PTR = 000000      M.QQAD= 000000
FF.RWD= 000001          F.FACC= 000043          GETNB = ****** GX       L.PTR1= 000004      M.QUAD= 000000
FF.RWF= 000006          F.FFBY= 000014          GNOME   003102R 025 L.SUBS= 000002      M.RAND= 000012
FF.SPC= 000004          F.FLAG= 000002          HASHC   005136R 025 MACHIN= 000106      M.RAVL= 000000
FHEADR  005012R 025 F.FLB = 000100              HASHTB  005140R 025 MAXCHN= 000014      M.RCIP= 000000
FHRET   005040R 025 F.FNAM= 000110              HSHMSK= 177601      MONBLK  003120R 025 M.RDCT= 000011
FILNTF= 104467          F.FNB = 000102          HSHSIZ= 000200      MONCVT  003132R 025 M.REVS= 000003
FIS   = 000000          F.FNDF= 000010          ICHAN   005050R 025 MONLEN  003130R 025 M.ROLL= 000000
FISCON= 046000          F.FTYP= 000116          ILFNAM= 104464      MONPTR  003122R 025 M.SATA= 000001
FIXPTR  003074R 025 F.FUN = 000010              ILLCHN= 104540      MONSFT  003126R 025 M.SCAN= 000011
FLAGS   004764R 025 F.FVER= 000120              ILLSW = 104565      MONTYP  003124R 025 M.SHAP= 000000
FLCBUF  003250R 025 F.HEAD= 000006              INDERR= 104410      MONVAL  003134R 025 M.SIGN= 000000
FLINE   005014R 025 F.HIBK= 000004              INNTMP  003250R 025 M.ABS = 000000      M.SUBR= 000001
FLOEXP  003270R 025 F.LBL = 000020              INNTM2  005016R 025 M.BOL = 000200      M.SYM = 000203
FLONEG  003272R 025 F.LINE= 000012              INPTR   000064R 025 M.BXAV= 000007      M.SYN = 000201
FLSWTB  000222R 025 F.LOC = 000004              INSNAM= 104462      M.BXCR= 000007      M.TRAN= 000012
FLTLEN= 000010          F.LSTC= 000004          IOERR = 104477      M.BXCT= 000007      M.$ABS= 000000
FNINSI  005044R 025 F.LUN = 000042              I.CNT = 177776      M.BXEX= 000007      M.$CEI= 000000
FORIGN  000032R 025 F.MBCT= 000054              I.DIM = 177772      M.BXFX= 000007      M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055          I.LEN = 000012      M.BXIO= 000007      M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OERR= 000062      M.BXLC= 000003      M.$NEG= 000000
FO.RD = 000001          F.MON = 000002          I.OLD = 177770      M.BXNC= 000007      M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001          I.PTR = 177774      M.BXNL= 000007      M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024          I.STR = 177766      M.BXPP= 000007      M.1REV= 000003
FPMODE  000042R 025 F.NREC= 000030              KILL  = ****** GX     M.BXPW= 000007      M.1SCN= 000011
FPP   = 000000          F.OPN = 000010          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FPPLEN= 000004          F.OVBS= 000030          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FREBLK  007400R 025 F.PREV= 000000              LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FUZZ    000142R 025 F.PRNT= 100000              LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FXPTR1  003076R 025 F.PROT= 010000              LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F.BBFS= 000062          F.RES = 000040          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.BDB = 000070          F.RET = 177772          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BKDN= 000026          F.R2  = 000014          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BKDS= 000020          F.R3  = 000016          LENTAB  000054R 025 M.GDWN= 000003      ND.$ES= 000020
F.BKEF= 000050          F.R4  = 000020          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
```

SYMBOL TABLE

```
ND.$N$= 000040          OPTMP3  005074R 025 ROUND   000010R 025 SYMTP   006272R 025 T.DIM = 000100
ND.$PN= 001400          OPTMP4  005076R 025 ROVER   003312R 025 SYMWRN= 000054     T.FUN = 010000
NEGSGN  005063R 025     OPTMP5  005100R 025 RSTS  = 000000     SYNERR= 104407     T.LIST= 000400
NINPT1  003302R 025     OPTMP6  005102R 025 RSX   = 000115     SYSER = ****** GX  T.LIT = 000020
NINPT2  003300R 025     OPTMP7  005104R 025 RSXERR= ****** GX  SYSERR= 104402     T.MON = 000010
NINSTK  003304R 025     OPTMP8  005106R 025 RTOP    005070R 025 SYSTAT  000000R 024 T.NIL = 000004
NINTMP  003310R 025     OPTMP9  005110R 025 RT11  = 000000     SY.BAD= 000001     T.NUM = 002000
NINTYP  003306R 025     OPTM10  005112R 025 R.BITS= 000003     SY.BRK= 100000     T.OPFN= 000001
NOTAVL= 104472          OPTM14  005026R 025 R.FIX = 000001     SY.BS = 000010     T.OTHR= 040000
NOTOPN= 104473          ORIGIN  000030R 025 R.LCT = 000002     SY.CLR= 000040     T.SOP = 000002
NPRT1   004762R 025     OUTPTR  000066R 025 R.LN  = 000002     SY.ERP= 000020     T.SPOP= 000040
NULL    007364R 025     OVROPR= 000000      R.LNUM= 000010     SY.NOU= 000100     T.TMP = 000200
NUMCHN= 000020          PAR$$$= 000000      R.NEXT= 000006     SY.OPN= 000002     T.VAL = 100000
NUMDFL  003230R 025     PATCH   007344R 025 R.OFF = 000000     SY.PMT= 001000     T.VAR = 001000
NUMEFL  003233R 025     PRECSN  000052R 025 R.PREV= 000004     SY.SRT= 000400     UPPER = 177400
NUMESN  003236R 025     RANDAC  003064R 025 R.PTR = 000000     SY.SVC= 000004     UTNAME  000344R 023
NUMFLG  003230R 025     RANDY   003062R 025 R.SEQ = 000003     SY.SVL= 000200     VALERR= 104413
NUMLEN  000062R 025     RANERR= 104411      R.VAR = 000002     S.AB  = 177776     VARBLE  007400R 025
NUMNEG  003232R 025     RARGBL  003204R 025 R6    =%000006     S.ASB = 100000     VARBLS  000000R 025
NUMQFL  003234R 025     RARGCV  003220R 025 R7    =%000007     S.BFHD= 000020     VERNUM  000000R 025
NUMRIG  003231R 025     RARGFL  003202R 025 SI    =%000005     S.BITS= 000002     VRSION= 000002
NUMWID  003226R 025     RARGLE  003216R 025 SIDAMN= 104433     S.CHR1= 000003     WIDTH   000072R 025
NUMZFL  003235R 025     RARGLN  003210R 025 SIFRAM  003314R 025 S.FATT= 000016     WROUND  000020R 025
NYIERR= 104432          RARGPT  003206R 025 SISTAK  006350R 025 S.FDB = 000140     WSDATE  000136R 025
N.BLSZ= 000036          RARGSF  003214R 025 SITOP   006346R 025 S.FLB = 040000     WSFULL= 104403
N.DESC= 000020          RARGTY  003212R 025 SITP    007144R 025 S.FNAM= 000006     WSNAME  000074R 025
N.DEV = 000032          RARGVA  003222R 025 SIWARN= 000200     S.FNB = 000036     WSTIME  000132R 025
N.DEVN= 000034          RBASE   005074R 025 SIZTAB  000044R 025 S.FNBW= 000017     $FIS  = ****** G
N.DID = 000024          RCOUNR  005112R 025 SI.BIT= 177774     S.FNTY= 000004     $FPP  = ****** G
N.DVNM= 000032          RCOUNT  005110R 025 SI.LEN= 177772     S.FTYP= 000002     $FPPLN= ****** G
N.EXT = 000014          REDCNT  005070R 025 SI.RET= 177772     S.LEN = 000002     $MACHN= ****** G
N.FID = 000000          REDIGN  005066R 025 SI.VAL= 177776     S.LINK= 177774     $RSTS = ****** G
N.FNAM= 000006          REDOFF  005072R 025 SKPSIZ  005134R 025 S.NFEN= 000020     $RSX  = ****** G
N.FTYP= 000014          REENTR  003060R 025 SPHOLD  000004R 024 S.TYPE= 140000     $RT11 = ****** G
N.FVER= 000016          RESBIT  003142R 025 SPINIT  000002R 024 S.VAL = 000000     .BOOLE= 000002
N.NAME= 000006          RESCVT  003152R 025 STKINC= 000774      TKDIMS  005072R 025 .CHARA= 000000
N.NEXT= 000022          RESERR= 104440      STOP  = 104420      TKFRAM  005066R 025 .CSI1 = ****** G
N.PPN = 000024          RESLEN  003150R 025 SUBCNT  003104R 025 TKLEFT  005070R 025 .CSI2 = ****** G
N.STAT= 000020          RESPTR  003140R 025 SUBDCT  003106R 025 TMODE   000403R 023 .DEF. = 000002
N.UNIT= 000034          RESSFT  003146R 025 SUBPTR  003110R 025 TMPTAG= 000166      .FLOAT= 000006
OBUFLN= 001000          RESTYP  003144R 025 SYMB1   005344R 025 TTYBUF  004140R 025 .FSRCB= ****** G
OCHAN   005052R 025     RESULT  003136R 025 SYMF    005340R 025 TTYLEN= 000622     .INTEG= 000004
OMODE   000402R 023     RESVAL  003154R 025 SYMPT1  005346R 025 TURKEY= 104436     .MOLUN  000404R 023
OPTMP0  005066R 025     RINC    005104R 025 SYMTAB  005352R 025 TYPE  = 000166     .PARSE= ****** GX
OPTMP1  005070R 025     RLTBAS  005066R 025 SYMTBF= 104404      T.BOL = 020000     ...GBL= 000000
OPTMP2  005072R 025     ROTINC  005070R 025 SYMTOP  005350R 025 T.CON = 004000     ...TPC= 000020

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
```

```
DECNAM  MACRO M1113  08-FEB-80 11:38  PAGE 46-4
SYMBOL TABLE

VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000374      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13044 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:42
DECNAM,DM1:[305,301]DECNAM/-SP=APLRX1,APLDEF,DECNAM,END
```

```
**
DD
DD
DD
DD
DD
DD
DD
DD
DD
DD
DD
DD
DD
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
```

```
**FILE**ID**DOT
```

```
DDDDDDDD        OOOOOO      TTTTTTTTTT
DDDDDDDD        OOOOOO      TTTTTTTTTT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT
DD      DD    OO      OO        TT          ....
DD      DD    OO      OO        TT          ....
DDDDDDDD        OOOOOO          TT          ....
DDDDDDDD        OOOOOO          TT          ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

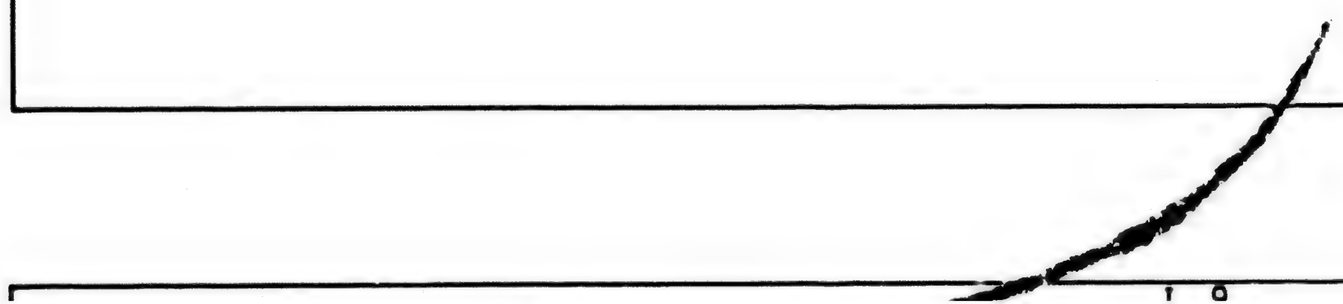
.MAIN. MACRO M1113 08-FEB-80 11:40
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    DOT              CHECK FOR INNER/OUTER PRODUCTS
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 1


      1  000000                RT11    =  0          ;NOT RT-11
      2  000000                RSTS    =  0          ;NOT RSTS/E
      3  000115                RSX     = 'M          ;RSX-11/M
      4
      5  000000                EIS     =  0          ;NO EIS
      6  000000                FIS     =  0          ;NO FIS
      7  000000                FPP     =  0          ;NO FPP
      8
      9  000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11  000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:40  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  DOT
                                .IDENT  \V02.03\
                                
                                .SBTTL  DOT             CHECK FOR INNER/OUTER PRODUCTS
                                
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine checks the syntax for inner/outer products.
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     DOT     :Jumped to from EXPR
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       calar op is on top of the SI.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       GETNXT  - Used to get next token in command string.
                                ;       VAL     - Used to get vals for inner/outer product
                                ;       INNERP  - For Inner Product.
                                ;       OUTERP  - For Outer Product.
                                ;
                                ; SIDE EFFECTS:
                                ;       The expression is computed, the result placed on the SI.
                                ;--
                                
 000000                         .PSECT  CODE
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 45
DOT             CHECK FOR INNER/OUTER PRODUCTS

     37 000000  032765  000002  177774  DOT::   BIT     #T.SOP,SI.BIT(SI);SCALAR OP ON SI?
     38 000006  001451                          BEQ     DOTSYN            ;NO, SYNTAX ERROR
     39 000010                                  CALL    GETNXT            ;GET NEXT TOKEN
     40 000014  032703  000002                  BIT     #T.SOP,R3         ;SCALAR OP?
     41 000020  001417                          BEQ     1$                ;NOPE, TRY FOR OUTER PRODUCT
     42 000022  012725  000132'                 MOV     #DOTSYN,(SI)+     ;PUSH OP ON SI
     43 000026  010325                          MOV     R3,(SI)+          ;PUSH BITS
     44 000030  010425                          MOV     R4,(SI)+          ;PUSH VALUE (OP NUMBER)
     45 000032                                  CALL    GETNXT            ;GET NEXT TOKEN
     46 000036                                  CALL    VAL               ;GET LEFT ARG
     47 000042  000433                          BR      DOTSYN            ;ERROR RETURN -->SYNTAX ERROR
     48 000044                                  PUSH    <R2,R3,R4>        ;SAVE SOME REGISTERS
     49 000052                                  CALL    D$INNR            ;PROCESS INNER PRODUCT
     50 000056  000420                          BR      2$                ;AND CYCLE
     51
     52                             ;TRY FOR OUTER PRODUCT
     53
     54 000060  032703  040000      1$:         BIT     #T.OTHR,R3        ;IS IT A CHAR?
     55 000064  001422                          BEQ     DOTSYN            ;NO
     56 000066  022704  000112                  CMP     #CHRSMO,R4        ;SMALL 'O'?
     57 000072  001017                          BNE     DOTSYN            ;NOPE
     58 000074                                  CALL    GETNXT            ;GET NEXT TOKEN
     59 000100                                  CALL    VAL               ;GET LEFT ARG
     60 000104  000412                          BR      DOTSYN            ;ERROR RETURN
     61 000106                                  PUSH    <R2,R3,R4>        ;SAVE SOME RESISTERS
     62 000114                                  CALL    M$OUTR            ;CALL OUTER PRODUCT
     63 000120                      2$:         POP     <R4,R3,R2>        ;RESTORE SOME REGISTERS
     64 000126  000137  000000G                 JMP     WORKER            ;AND CONTINUE
     65
     66 000132  104407              DOTSYN:     SYNERR                    ;SYNTAX ERROR
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 47
DOT             CHECK FOR INNER/OUTER PRODUCTS

     1          000001                          .END
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 47-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073   C.BLSZ= 000140   D.CTLM= 000000   EDIFLG  005062R  025
AC1    =%000001          CHRLPR= 000072   C.BUFF= 177760   D.DEAL= 000003   EDIT   = 000003
AC2    =%000002          CHRMNS= 000137   C.BUFL= 177762   D.DECD= 000002   EIS    = 000000
AC3    =%000003          CHRMUL= 000075   C.CCNT= 177766   D.DIV = 000000   ELESIZ  005132R  025
AC4    =%000004          CHRNEG= 000100   C.CPTR= 177764   D.DMD = 000010   ENDCOR  000004R  025
AC5    =%000005          CHRPDL= 000271   C.CZ   = 001000   D.DQAD= 177777   EOF    = 104475
ALLINT  003274R  025     CHRPLU= 000055   C.DIG  = 000040   D.DROP= 000002   ERRMAX= 000166
ARYRET  003276R  025     CHRPWR= 000120   C.FASW= 020000   D.ENCD= 000002   ERRNUM  004772R  025
ASCQUO= 000042           CHRQST= 000121   C.FLEN= 000010   D.EQ   = 000000   ERRPTR  004770R  025
A.ALEN= 000004           CHRQUO= 000113   C.FLSH= 004000   D.EXPD= 000002   ERRO    004774R  025
A.ARG  = 000040          CHRRBR= 000047   C.FRC  = 000001   D.FGET= 000005   EXEBUF  005054R  025
A.ASB  = 000200          CHRRPR= 000042   C.IDEN= 000010   D.FMT  = 000004   E$CT    005056R  025
A.BITS= 000003           CHRSEM= 000074   C.NAME= 000102   D.FPNT= 000005   E$PT    005060R  025
A.CON  = 000020          CHRSMO= 000112   C.NOBF= 000400   D.FPUT= 000005   FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040   C.NUM  = 000020   D.GE   = 000000   FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131   C.OP   = 000004   D.GT   = 000000   FA.DLK= 001000
A.NUM  = 000006          CHR.BS= 000010   C.OPN  = 010000   D.IBM  = 000002   FA.ENB= 100000
A.PTR  = 000000          CHR.CC= 000003   C.RDLN= 177776   D.IND  = 000002   FA.EXC= 002000
A.TMP  = 000010          CHR.CR= 000015   C.REWT= 002000   D.INNR= 000013   FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032   C.RND  = 000010   D.INTR= 000006   FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046   C.SBIT= 177756   D.LE   = 000000   FA.NSR= 000200
BLZBUB  005116R  025     CHR.LF= 000012   C.SPFX= 000100   D.LOG  = 000001   FA.POS= 010000
BRSELF  005046R  025     CHR.LT= 000043   C.SPGL= 000200   D.LT   = 000000   FA.RD  = 000001
CCNT    005072R  025     CHR.SI= 000017   C.SPLX= 000001   D.MAX  = 000000   FA.RWD= 004000
CHANEX  003060R  025     CHR.SL= 000057   C.SPRS= 000002   D.MEM  = 000002   FA.SEQ= 040000
CHAN00  000256R  025     CHR.SO= 000016   C.SRC  = 177774   D.MIN  = 000000   FA.SHR= 000040
CHAN01  000330R  025     CHR.TB= 000011   C.SRCP= 000024   D.MUL  = 000000   FA.TMP= 000020
CHAN02  000512R  025     CHR.UL= 000106   C.TT   = 000004   D.NE   = 000000   FA.WCK= 020000
CHAN03  000674R  025     CHR..A= 000141   C.WIDC= 177770   D.NN   = 000000   FA.WRT= 000002
CHAN04  001056R  025     CHR..E= 000145   C.WIDR= 177772   D.NR   = 000000   FD.BLK= 000010
CHAN05  001240R  025     CHR..S= 000163   DEBUG  = 000000   D.OR   = 000000   FD.CCL= 000002
CHAN06  001422R  025     CHR..T= 000164   DEFERR= 104405   D.PLUS= 000000   FD.COM= 020000
CHAN07  001604R  025     CHR..U= 000165   DEVIL   005114R  025 D.PSUB= 000006   FD.CR  = 000002
CHAN08  001766R  025     CHR..Z= 000172   DEVNTF= 104470   D.PWR  = 000001   FD.DIR= 000010
CHAN09  002150R  025     CHR..0= 000060   DIGITS  000006R  025 D.QQAD= 000000   FD.FTN= 000001
CHAN10  002332R  025     CHR..1= 000061   DIMAX  = 000020   D.QUAD= 000000   FD.F11= 040000
CHAN11  002514R  025     CH.AND= 000001   DIMBLK  003112R  025 D.RES  = 000000   FD.INS= 000010
CHAN12  002676R  025     CMDBUF  003250R  025 DIMPTR  003114R  025 D.RHO  = 000000   FD.ISP= 002000
CHAN13  000022R  023     CMDMAX= 000010   DIMTYP  003116R  025 D.ROT  = 000003   FD.MNT= 100000
CHAN14  000204R  023     CNTSIZ  005130R  025 DIRFUL= 104466   D.SCON= 000006   FD.OSP= 004000
CHNOPN= 104471           COMBIN  005076R  025 DOMERR= 104417   D.SDIF= 000006   FD.PLC= 000004
CHRAND= 000051           COMCNT  005126R  025 DOT     000000RG  020 D.SUB  = 000000   FD.PRN= 000004
CHRBEL= 000007           COMERR= 104425   DOTSYN  000132R  020 D.SUBS= 000006   FD.PSE= 010000
CHRBKR= 000133           COMPTR  005122R  025 DPTERR= 104416   D.TAKE= 000002   FD.RAH= 000001
CHRBOL= 000200           COMRTN  003156R  025 DYDCNT  003224R  025 D.TRAN= 000004   FD.RAN= 000002
CHRBOX= 000114           COMSKP  005124R  025 D$INNR= ****** GX   D.UNON= 000006   FD.REC= 000001
CHRCMA= 000054           CORPTR  005120R  025 D.AND  = 000000   D.$MAX= 000000   FD.RWM= 000001
CHRCOL= 000076           CS.ASS= 000200   D.BXCT= 000007   D.$MIN= 000000   FD.SDI= 000020
CHRDEL= 000107           CS.BOO= 000010   D.BXIO= 000007   D.$MUL= 000000   FD.SQD= 000040
CHRDIV= 000053           CS.FLO= 000020   D.BXNL= 000007   D.$PLU= 000000   FD.TTY= 000004
CHRDLR= 000176           CS.IDN= 000040   D.BXPP= 000007   D.$RES= 000000   FD.WBH= 000002
CHRDLT= 000110           CS.ONE= 000100   D.BXPW= 000007   D.$SUB= 000000   FF.CHR= 000005
CHRDOT= 000056           CTEPS   000152R  025 D.BXRL= 000007   D.1COM= 000001   FF.NV  = 000003
CHREP  = 000105          CTMAX   000212R  025 D.CIRC= 000001   D.1EXP= 000002   FF.POE= 000002
CHRERR= 104431           CTRMAX  000172R  025 D.COMB= 000001   D.1ROT= 000003   FF.RWD= 000001
CHRFIL= 000177           C.ALOC= 000162   D.COMP= 000001   EDICHR  005063R  025 FF.RWF= 000006
CHRFRO= 000135           C.BLOK= 000064   D.CONT= 000006   EDICNT  005064R  025 FF.SPC= 000004
```

DOT MACRO M1113 08-FEB-80 11:40 PAGE 47-2
SYMBOL TABLE

```
FHEADR  005012R      025 F.FLB = 000100          HSHMSK= 177601          MONCVT  003132R      025 M.RDCT= 000011
FHRET   005040R      025 F.FNAM= 000110          HSHSIZ= 000200          MONLEN  003130R      025 M.REVS= 000003
FILNTF= 104467           F.FNB = 000102          ICHAN   005050R  025   MONPTR  003122R      025 M.ROLL= 000000
FIS   = 000000           F.FNDF= 000010          ILFNAM= 104464          MONSFT  003126R      025 M.SATA= 000001
FISCON= 046000           F.FTYP= 000116          ILLCHN= 104540          MONTYP  003124R      025 M.SCAN= 000011
FIXPTR  003074R      025 F.FUN = 000010          ILLSW = 104565          MONVAL  003134R      025 M.SHAP= 000000
FLAGS   004764R      025 F.FVER= 000120          INDERR= 104410          M$OUTR= ****** GX        M.SIGN= 000000
FLCBUF  003250R      025 F.HEAD= 000006          INNTMP  003250R  025   M.ABS = 000000            M.SUBR= 000001
FLINE   005014R      025 F.HIBK= 000004          INNTM2  005016R  025   M.BOL = 000200            M.SYM = 000203
FLOEXP  003270R      025 F.LBL = 000020          INPTR   000064R  025   M.BXAV= 000007            M.SYN = 000201
FLONEG  003272R      025 F.LINE= 000012          INSNAM= 104462          M.BXCR= 000007            M.TRAN= 000012
FLSWTB  000222R      025 F.LOC = 000004          IOERR = 104477          M.BXCT= 000007            M.$ABS= 000000
FLTLEN= 000010           F.LSTC= 000004          I.CNT = 177776          M.BXEX= 000007            M.$CEI= 000000
FNINSI  005044R      025 F.LUN = 000042          I.DIM = 177772          M.BXFX= 000007            M.$FLR= 000000
FORIGN  000032R      025 F.MBCT= 000054          I.LEN = 000012          M.BXIO= 000007            M.$IDN= 000000
FO.APD= 000106           F.MBC1= 000055          I.OERR= 000062          M.BXLC= 000003            M.$NEG= 000000
FO.MFY= 000002           F.MBFG= 000056          I.OLD = 177770          M.BXNC= 000007            M.$SIG= 000000
FO.RD = 000001           F.MON = 000002          I.PTR = 177774          M.BXNL= 000007            M.1RDT= 000011
FO.UPD= 000006           F.NIL = 000001          I.STR = 177766          M.BXPP= 000007            M.1REV= 000003
FO.WRT= 000016           F.NRBD= 000024          LABERR= 104406          M.BXPW= 000007            M.1SCN= 000011
FPMODE  000042R      025 F.NREC= 000030          LABLPT  005042R  025   M.BXRL= 000007            NAMBUF  003240R      025
FPP   = 000000           F.OPN = 000010          LARGBL  003162R  025   M.BXTS= 000007            NB.DEV= 000200
FPPLEN= 000004           F.OVBS= 000030          LARGCV  003176R  025   M.BXWA= 000007            NB.DIR= 000100
FREBLK  007400R      025 F.PREV= 000000          LARGFL  003160R  025   M.CEI = 177777            NB.NAM= 000004
FUZZ    000142R      025 F.PRNT= 100000          LARGLE  003174R  025   M.CHR = 000202            NB.SD1= 000400
FXPTR1  003076R      025 F.PROT= 010000          LARGLN  003166R  025   M.CMT = 000206            NB.SD2= 001000
F$EVAL= 000100           F.RACC= 000016          LARGPT  003164R  025   M.DOUT= 000006            NB.SNM= 000040
F$EVI = 000200           F.RATT= 000001          LARGSF  003172R  025   M.DQAD= 000006            NB.STP= 000020
F$EXEC= 000400           F.RCNM= 000034          LARGTY  003170R  025   M.ENCD= 000004            NB.SVR= 000010
F$FUNC= 001000           F.RCTL= 000017          LARGVA  003200R  025   M.EXE = 000000            NB.TYP= 000002
F.ACTL= 000076           F.READ= 004000          LBINC   005100R  025   M.EXP = 000001            NB.VER= 000001
F.ALOC= 000040           F.REDF= 000040          LBUFF   003316R  025   M.FACT= 000001            ND.DEV= 000200
F.BBFS= 000062           F.RES = 000040          LBUFLN= 000621          M.FIL = 000177            ND.EXT= 000002
F.BDB = 000070           F.RET = 177772          LCOUNT  005106R  025   M.FLR = 177777            ND.NAM= 000004
F.BGBC= 000057           F.RSIZ= 000002          LENERR= 104412          M.FMT = 000004            ND.PPN= 000100
F.BITS= 177774           F.RTYP= 000000          LENTAB  000054R  025   M.GDUP= 000003            ND.$EX= 000000
F.BKDN= 000026           F.R2  = 000014          LEXPTR  003100R  025   M.GDWN= 000003            ND.$E$= 000020
F.BKDS= 000020           F.R3  = 000016          LFTOP   005066R  025   M.GETI= 000006            ND.$NM= 000000
F.BKEF= 000050           F.R4  = 000020          LINC    005102R  025   M.IBM = 000003            ND.$N$= 000040
F.BKP1= 000051           F.SEQN= 000100          LINE    005016R  025   M.IDN = 000000            ND.$PN= 001400
F.BKST= 000024           F.SPDV= 000072          LININC  005026R  025   M.INSP= 000006            NEGSGN  005063R      025
F.BKVB= 000064           F.SPUN= 000074          LISIZE= 000020          M.IOTA= 000000            NINPT1  003302R      025
F.BR  = 000020           F.STBK= 000036          LITOP   005072R  025   M.LIT = 000205            NINPT2  003300R      025
F.CHR = 000075           F.STK = 000022          LNLNG = 104541          M.LN  = 000001            NINSTK  003304R      025
F.CNTG= 000034           F.STOP= 000004          LOCLCT  005036R  025   M.MMD = 000010            NINTMP  003310R      025
F.CODE= 000010           F.STP = 000001          LOWCOR  000002R  025   M.NEG = 000000            NINTYP  003306R      025
F.DFNB= 000046           F.SUSP= 000002          LSTERP  004766R  025   M.NOT = 000000            NOTAVL= 104472
F.DSPT= 000044           F.TRC = 000002          LWIDTH  000070R  025   M.NUB = 000006            NOTOPN= 104473
F.DVNM= 000134           F.UNIT= 000136          L.BITS= 000003          M.NUM = 000204            NPRT1   004762R      025
F.EFBK= 000010           F.URBD= 000020          L.CNT = 000002          M.OUTR= 000012            NULL    007364R      025
F.EFN = 000050           F.VAL = 177776          L.PRNT= 000001          M.PITM= 000000            NUMCHN= 000020
F.END = 002000           F.VBN = 000064          L.PTR = 000000          M.PRDS= 000006            NUMDFL  003230R      025
F.EOBB= 000032           F.VBSZ= 000060          L.PTR1= 000004          M.QQAD= 000000            NUMEFL  003233R      025
F.ERR = 000052           GETNXT= ****** GX       L.SUBS= 000002          M.QUAD= 000000            NUMESN  003236R      025
F.FACC= 000043           GNOME   003102R  025 MACHIN= 000106            M.RAND= 000012            NUMFLG  003230R      025
F.FFBY= 000014           HASHC   005136R  025 MAXCHN= 000014            M.RAVL= 000000            NUMLEN  000062R      025
F.FLAG= 000002           HASHTB  005140R  025 MONBLK  003120R  025   M.RCIP= 000000            NUMNEG  003232R      025
```

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 47-3
SYMBOL TABLE

NUMQFL  003234R     025 RANDY   003062R     025 R.OFF  = 000000          SY.CLR= 000040          T.MON  = 000010
NUMRIG  003231R     025 RANERR= 104411          R.PREV= 000004          SY.ERP= 000020          T.NIL  = 000004
NUMWID  003226R     025 RARGBL  003204R     025 R.PTR  = 000000          SY.NOU= 000100          T.NUM  = 002000
NUMZFL  003235R     025 RARGCV  003220R     025 R.SEQ  = 000003          SY.OPN= 000002          T.OPFN= 000001
NYIERR= 104432          RARGFL  003202R     025 R.VAR  = 000002          SY.PMT= 001000          T.OTHR= 040000
N.BLSZ= 000036          RARGLE  003216R     025 R6     =%000006          SY.SRT= 000400          T.SOP  = 000002
N.DESC= 000020          RARGLN  003210R     025 R7     =%000007          SY.SVC= 000004          T.SPOP= 000040
N.DEV  = 000032         RARGPT  003206R     025 SI     =%000005          SY.SVL= 000200          T.TMP  = 000200
N.DEVN= 000034          RARGSF  003214R     025 SIDAMN= 104433           S.AB   = 177776          T.VAL  = 100000
N.DID  = 000024         RARGTY  003212R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAR  = 001000
N.DVNM= 000032          RARGVA  003222R     025 SISTAK  006350R     025 S.BFHD= 000020          UPPER  = 177400
N.EXT  = 000014         RBASE   005074R     025 SITOP   006346R     025 S.BITS= 000002          UTNAME  000344R     023
N.FID  = 000000         RCOUNR  005112R     025 SITP    007144R     025 S.CHR1= 000003          VAL    = ****** GX
N.FNAM= 000006          RCOUNT  005110R     025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.FTYP= 000014          REDCNT  005070R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLE  007400R     025
N.FVER= 000016          REDIGN  005066R     025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R     025
N.NAME= 000006          REDOFF  005072R     025 SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R     025
N.NEXT= 000022          REENTR  003060R     025 SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
N.PPN  = 000024         RESBIT  003142R     025 SI.VAL= 177776           S.FNBW= 000017          WIDTH   000072R     025
N.STAT= 000020          RESCVT  003152R     025 SKPSIZ  005134R     025 S.FNTY= 000004          WORKER= ****** GX
N.UNIT= 000034          RESERR= 104440          SPHOLD  000004R     024 S.FTYP= 000002          WROUND  000020R     025
OBUFLN= 001000          RESLEN  003150R     025 SPINIT  000002R     024 S.LEN  = 000002          WSDATE  000136R     025
OCHAN   005052R     025 RESPTR  003140R     025 STKINC= 000774           S.LINK= 177774          WSFULL= 104403
OMODE   000402R     023 RESSFT  003146R     025 STOP   = 104420           S.NFEN= 000020          WSNAME  000074R     025
OPTMP0  005066R     025 RESTYP  003144R     025 SUBCNT  003104R     025 S.TYPE= 140000          WSTIME  000132R     025
OPTMP1  005070R     025 RESULT  003136R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FIS   = ****** G
OPTMP2  005072R     025 RESVAL  003154R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPP   = ****** G
OPTMP3  005074R     025 RINC    005104R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $FPPLN= ****** G
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMF    005340R     025 TKLEFT  005070R     025 $MACHN= ****** G
OPTMP5  005100R     025 ROTINC  005070R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSTS  = ****** G
OPTMP6  005102R     025 ROUND   000010R     025 SYMTAB  005352R     025 TMPTAG= 000166          $RSX   = ****** G
OPTMP7  005104R     025 ROVER   003312R     025 SYMTBF= 104404           TTYBUF  004140R     025 $RT11  = ****** G
OPTMP8  005106R     025 RSTS   = 000000          SYMTOP  005350R     025 TTYLEN= 000622          .BOOLE= 000002
OPTMP9  005110R     025 RSX    = 000115          SYMTP   006272R     025 TURKEY= 104436          .CHARA= 000000
OPTM10  005112R     025 RTOP    005070R     025 SYMWRN= 000054           TYPE   = 000166          .DEF.  = 000002
OPTM14  005026R     025 RT11   = 000000          SYNERR= 104407           T.BOL  = 020000          .FLOAT= 000006
ORIGIN  000030R     025 R.BITS= 000003           SYSERR= 104402           T.CON  = 004000          .FSRCB= ****** G
OUTPTR  000066R     025 R.FIX  = 000001          SYSTAT  000000R     024 T.DIM  = 000100          .INTEG= 000004
OVROPR= 000000          R.LCT  = 000002          SY.BAD= 000001           T.FUN  = 010000          .MOLUN  000404R     023
PATCH   007344R     025 R.LN   = 000002          SY.BRK= 100000           T.LIST= 000400          ...GBL= 000000
PRECSN  000052R     025 R.LNUM= 000010           SY.BS  = 000010          T.LIT  = 000020          ...TPC= 000020
RANDAC  003064R     025 R.NEXT= 000006

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

B 10

```
DOT     MACRO M1113  08-FEB-80 11:40  PAGE 47-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000134      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12929 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:54
DOT,DM1:[305,301]DOT/-SP=APLRX1,APLDEF,DOT,END
```

C 10

```
**FILE**ID**DROPC
```

```
DDDDDDDD      RRRRRRRR        000000     PPPPPPPP        CCCCCCCC
DDDDDDDD      RRRRRRRR        000000     PPPPPPPP        CCCCCCCC
DD      DD    RR      RR    00      00   PP      PP    CC
DD      DD    RR      RR    00      00   PP      PP    CC
DD      DD    RR      RR    00      00   PP      PP    CC
DD      DD    RR      RR    00      00   PP      PP    CC
DD      DD    RRRRRRRR      00      00   PPPPPPPP      CC
DD      DD    RRRRRRRR      00      00   PPPPPPPP      CC
DD      DD    RR  RR        00      00   PP            CC
DD      DD    RR  RR        00      00   PP            CC
DD      DD    RR    RR      00      00   PP            CC                ....
DD      DD    RR    RR      00      00   PP            CC                ....
DDDDDDDD      RR      RR      000000     PP              CCCCCCCC        ....
DDDDDDDD      RR      RR      000000     PP              CCCCCCCC        ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 11:42
TABLE OF CONTENTS

2- 1 APLDEF
2- 8 A P L
2- 9
2- 10 A PROGRAMMING LANGUAGE
2- 11
2- 12 STEVE POULSEN & DAVE VANN
2- 13 11-Jul-76
2- 14 ROBERT GOTTLIEB & PETER HAYNES
2- 15 2-Feb-79
2- 16
2- 17 V2
2- 18
2- 19
2- 20 COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
2- 21 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
2- 22
2- 23
2- 24 THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
2- 25 COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
2- 26 COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
2- 27 BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
2- 28 USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
2- 29 TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
2- 30 DEC.
2- 31
2- 32 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
2- 33 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
2- 34 CORPORATION.
2- 35
2- 36 DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
2- 37 SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
2- 38
2- 39
3- 46 DEFINITIONS ASSEMBLY PARAMETERS
4- 139 DEFINITIONS OPERATOR OVERLAYS
5- 268 DEFINITIONS REGISTERS
6- 331 DEFINITIONS MEMORY ORGANIZATION
7- 377 DEFINITIONS ORDERING PSECTS
8- 426 DEFINITIONS SI PARSE FRAMES
9- 474 DEFINITIONS TOKEN DESCRIPTION BITS
10- 515 DEFINITIONS SI EXECUTION FRAMES
11- 580 DEFINITIONS SPECIAL PARSE CHARACTERS
12- 607 DEFINITIONS CHRTAB BITS
[illegible] 624 DEFINITIONS SYSTEM STATUS BITS
[illegible] 640 DEFINITIONS F-FRAMES
[illegible] 724 DEFINITIONS FUNCTION LINE BLOCKS
[illegible] 807 DEFINITIONS ARRAY STORAGE BLOCK
[illegible] 869 DEFINITIONS LIST STORAGE BLOCK
[illegible] 911 DEFINITIONS INDEXING STACK FRAME
[illegible] 940 DEFINITIONS SYMBOL TABLE ENTRY
[illegible] 969 DEFINITIONS CHANNEL BLOCK
[illegible] 1085 DEFINITIONS RSX DATA
[illegible] 1110 DEFINITIONS NAME BLOCKS
33- 1612 DEFINITIONS MACROS
35- 1905 DEFINITIONS APL CHARACTERS
36- 1967 DEFINITIONS ERRORS
44- 4 COMMANDS DROP

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 1


      1  000000                RT11    =  0          ;NOT RT-11
      2  000000                RSTS    =  0          ;NOT RSTS/E
      3  000115                RSX     =  'M         ;RSX-11/M
      4
      5  000000                EIS     =  0          ;NO EIS
      6  000000                FIS     =  0          ;NO FIS
      7  000000                FPP     =  0          ;NO FPP
      8
      9  000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11  000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:42  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  DROPC
                                        .IDENT  \V02.03\
                                        .SBTTL  COMMANDS                DROP
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       DELETES A FILE BY NAME
                                        ;--
000000                                          .PSECT  CODE
000000                                  DROPC::
000000  012700  000204'                         MOV     #CHAN14,R0      ;WORK FDB
000004  012701  000306'                         MOV     #CHAN14+C.NAME,R1;NAME BLOCK
000010  012702  000000G                         MOV     #DFLTSV,R2      ;DEFAULT STUFF
000014                                          CALL    DECNAM          ;DECODE NAME, W/ STARS
000020  032761  001470  000020                  BIT     #ND.$N$!ND.$NM!ND.$E$!ND.$EX!ND.$PN!NB.SVR,N.DESC(R1)
000026  001004                                  BNE     1$              ;YUP...
000030                                          CALL    .DLFNB          ;DELETE THE FILE
000034                                          CALLR   RSXERR          ;CHECK FOR RSX-11 ERRORS
000040                                  1$:     PUSH    R0              ;Save R0
000042                                          MPRINT  FILESD          ;" FILES DELETED:"
000056                                          POP     R0              ;Restore it
000060                                  2$:     CALL    .FIND           ;FIND NEXT ELEMENT IN LIST
000064  103431                                  BCS     9$              ;FINISHED?
000066                                          PUSH    <N.NEXT(R1),N.DESC(R1)>
                                                                        ;Save stuff for next lookup
000076  005061  000020                          CLR     N.DESC(R1)      ;Clear these
000102  005061  000022                          CLR     N.NEXT(R1)      ;   fields for delete
000106                                          CALL    .DLFNB          ;DELETE IT
000112                                          CALL    RSXERR          ;CHECK FOR ERRORS
000116  010104                                  MOV     R1,R4           ;Point to the name block
000120  011661  000020                          MOV     (SP),N.DESC(R1) ;Restore Stat bits
000124  052761  000007  000020                  BIS     #NB.VER!NB.TYP!NB.NAM,N.DESC(R1)
                                                                        ;Force the Name, Extension, and Version
000132                                          CALL    WSPRT           ;Print the name
000136                                          POP     <N.DESC(R1),N.NEXT(R1)>
                                                                        ;Restore stuff
000146  000744                                  BR      2$              ;AND LOOP
000150                                  9$:     CALL    NEWDT           ;GET CURRENT DATE, TIME
000154                                          CALLR   NEWLIN          ;End it w/ a <CR><LF>
000000                                          .PSECT  TABLES
000000                                  FILESD:
000000      144     162     157                 .ASCIZ  /dropped>/<CHR.CR><CHR.LF>
000003      160     160     145
000006      144     076     015
000011      012     000
                                                .EVEN
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 45
COMMANDS                 DROP

      1          000001                              .END
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 45-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073          C.BLSZ= 000140          D.CTLM= 000000          EDIFLG  005062R     025
AC1    =%000001          CHRLPR= 000072          C.BUFF= 177760          D.DEAL= 000003          EDIT   = 000003
AC2    =%000002          CHRMNS= 000137          C.BUFL= 177762          D.DECD= 000002          EIS    = 000000
AC3    =%000003          CHRMUL= 000075          C.CCNT= 177766          D.DIV = 000000          ELESIZ  005132R     025
AC4    =%000004          CHRNEG= 000100          C.CPTR= 177764          D.DMD = 000010          ENDCOR  000004R     025
AC5    =%000005          CHRPDL= 000271          C.CZ  = 001000          D.DQAD= 177777          EOF    = 104475
ALLINT  003274R     025  CHRPLU= 000055          C.DIG = 000040          D.DROP= 000002          ERRMAX= 000166
ARYRET  003276R     025  CHRPWR= 000120          C.FASW= 020000          D.ENCD= 000002          ERRNUM  004772R     025
ASCQUO= 000042           CHRQST= 000121          C.FLEN= 000010          D.EQ  = 000000          ERRPTR  004770R     025
A.ALEN= 000004           CHRQUO= 000113          C.FLSH= 004000          D.EXPD= 000002          ERR0    004774R     025
A.ARG = 000040           CHRRBR= 000047          C.FRC = 000001          D.FGET= 000005          EXEBUF  005054R     025
A.ASB = 000200           CHRRPR= 000042          C.IDEN= 000010          D.FMT = 000004          E$CT    005056R     025
A.BITS= 000003           CHRSEM= 000074          C.NAME= 000102          D.FPNT= 000005          E$PT    005060R     025
A.CON = 000020           CHRSMO= 000112          C.NOBF= 000400          D.FPUT= 000005          FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040          C.NUM = 000020          D.GE  = 000000          FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131          C.OP  = 000004          D.GT  = 000000          FA.DLK= 001000
A.NUM = 000006           CHR.BS= 000010          C.OPN = 010000          D.IBM = 000002          FA.ENB= 100000
A.PTR = 000000           CHR.CC= 000003          C.RDLN= 177776          D.IND = 000002          FA.EXC= 002000
A.TMP = 000010           CHR.CR= 000015          C.REWT= 002000          D.INNR= 000013          FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032          C.RND = 000010          D.INTR= 000006          FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046          C.SBIT= 177756          D.LE  = 000000          FA.NSR= 000200
BLZBUB  005116R     025  CHR.LF= 000012          C.SPFX= 000100          D.LOG = 000001          FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043          C.SPGL= 000200          D.LT  = 000000          FA.RD = 000001
CCNT    005072R     025  CHR.SI= 000017          C.SPLX= 000001          D.MAX = 000000          FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057          C.SPRS= 000002          D.MEM = 000002          FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016          C.SRC = 177774          D.MIN = 000000          FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011          C.SRCP= 000024          D.MUL = 000000          FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106          C.TT  = 000004          D.NE  = 000000          FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141          C.WIDC= 177770          D.NN  = 000000          FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145          C.WIDR= 177772          D.NR  = 000000          FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163          DEBUG = 000000          D.OR  = 000000          FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164          DECNAM= ****** GX       D.PLUS= 000000          FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165          DEFERR= 104405          D.PSUB= 000006          FD.CR = 000002
CHAN08  001766R     025  CHR..Z= 000172          DEVIL   005114R     025 D.PWR = 000001          FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060          DEVNTF= 104470          D.QQAD= 000000          FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061          DFLTSV= ****** GX       D.QUAD= 000000          FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001          DIGITS  000006R     025 D.RES = 000000          FD.INS= 000010
CHAN12  002676R     025  CMDBUF  003250R     025 DIMAX = 000020          D.RHO = 000000          FD.ISP= 002000
CHAN13  000022R     023  CMDMAX= 000010          DIMBLK  003112R     025 D.ROT = 000003          FD.MNT= 100000
CHAN14  000204R     023  CNTSIZ  005130R     025 DIMPTR  003114R     025 D.SCON= 000006          FD.OSP= 004000
CHNOPN= 104471           COMBIN  005076R     025 DIMTYP  003116R     025 D.SDIF= 000006          FD.PLC= 000004
CHRAND= 000051           COMCNT  005126R     025 DIRFUL= 104466          D.SUB = 000000          FD.PRN= 000004
CHRBEL= 000007           COMERR= 104425          DOMERR= 104417          D.SUBS= 000006          FD.PSE= 010000
CHRBKR= 000133           COMPTR  005122R     025 DPTERR= 104416          D.TAKE= 000002          FD.RAH= 000001
CHRBOL= 000200           COMRTN  003156R     025 DROPC   000000RG  020 D.TRAN= 000004          FD.RAN= 000002
CHRBOX= 000114           COMSKP  005124R     025 DYDCNT  003224R     025 D.UNON= 000006          FD.REC= 000001
CHRCMA= 000054           CORPTR  005120R     025 D.AND = 000000          D.$MAX= 000000          FD.RWM= 000001
CHRCOL= 000076           CS.ASS= 000200          D.BXCT= 000007          D.$MIN= 000000          FD.SDI= 000020
CHRDEL= 000107           CS.BOO= 000010          D.BXIO= 000007          D.$MUL= 000000          FD.SQD= 000040
CHRDIV= 000053           CS.FLO= 000020          D.BXNL= 000007          D.$PLU= 000000          FD.TTY= 000004
CHRDLR= 000176           CS.IDN= 000040          D.BXPP= 000007          D.$RES= 000000          FD.WBH= 000002
CHRDLT= 000110           CS.ONE= 000100          D.BXPW= 000007          D.$SUB= 000000          FF.CHR= 000005
CHRDOT= 000056           CTEPS   000152R     025 D.BXRL= 000007          D.1COM= 000001          FF.NV = 000003
CHREP = 000105           CTMAX   000212R     025 D.CIRC= 000001          D.1EXP= 000002          FF.POE= 000002
CHRERR= 104431           CTRMAX  000172R     025 D.COMB= 000001          D.1ROT= 000003          FF.RWD= 000001
CHRFIL= 000177           C.ALOC= 000162          D.COMP= 000001          EDICHR  005063R     025 FF.RWF= 000006
CHRFRO= 000135           C.BLOK= 000064          D.CONT= 000006          EDICNT  005064R     025 FF.SPC= 000004
```

J 10

SYMBOL TABLE

```
FHEADR  005012R     025 F.FLAG= 000002      HSHMSK= 177601          MONCVT  003132R     025 M.REVS= 000003
FHRET   005040R     025 F.FLB = 000100      HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000
FILESD  000000R     002 F.FNAM= 000110      ICHAN   005050R     025 MONPTR  003122R     025 M.SATA= 000001
FILNTF= 104467          F.FNB = 000102      ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FIS    = 000000         F.FNDF= 000010      ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116      ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FIXPTR  003074R     025 F.FUN = 000010      INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R     025 F.FVER= 000120      INNTMP  003250R     025 M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R     025 F.HEAD= 000006      INNTM2  005016R     025 M.BXAV= 000007          M.SYN = 000201
FLINE   005014R     025 F.HIBK= 000004      INPTR   000064R     025 M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R     025 F.LBL = 000020      INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R     025 F.LINE= 000012      IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R     025 F.LOC = 000004      I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004      I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R     025 F.LUN = 000042      I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055      I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056      I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002      I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FO.WRT= 000016          F.NRBD= 000024      LABLPT  005042R     025 M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R     025 F.NREC= 000030      LARGBL  003162R     025 M.BXWA= 000007          NB.DIR= 000100
FPP    = 000000         F.OPN = 000010      LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030      LARGFL  003160R     025 M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R     025 F.PREV= 000000      LARGLE  003174R     025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R     025 F.PRNT= 100000      LARGLN  003166R     025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R     025 F.PROT= 010000      LARGPT  003164R     025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016      LARGSF  003172R     025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001      LARGTY  003170R     025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034      LARGVA  003200R     025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017      LBINC   005100R     025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000      LBUFF   003316R     025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040      LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040      LCOUNT  005106R     025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772      LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002      LENTAB  000054R     025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000      LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014      LFTOP   005066R     025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016      LINC    005102R     025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020      LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BKP1= 000051          F.SEQN= 000100      LININC  005026R     025 M.IOTA= 000000          NEWDT = ****** GX
F.BKST= 000024          F.SPDV= 000072      LISIZE= 000020          M.LIT = 000205          NEWLIN= ****** GX
F.BKVB= 000064          F.SPUN= 000074      LITOP   005072R     025 M.LN  = 000001          NINPT1  003302R     025
F.BR   = 000020         F.STBK= 000036      LNLNG = 104541          M.MMD = 000010          NINPT2  003300R     025
F.CHR = 000075          F.STK = 000022      LOCLCT  005036R     025 M.NEG = 000000          NINSTK  003304R     025
F.CNTG= 000034          F.STOP= 000004      LOWCOR  000002R     025 M.NOT = 000000          NINTMP  003310R     025
F.CODE= 000010          F.STP = 000001      LSTERP  004766R     025 M.NUB = 000006          NINTYP  003306R     025
F.DFNB= 000046          F.SUSP= 000002      LWIDTH  000070R     025 M.NUM = 000204          NOTAVL= 104472
F.DSPT= 000044          F.TRC = 000002      L.BITS= 000003          M.OUTR= 000012          NOTOPN= 104473
F.DVNM= 000134          F.UNIT= 000136      L.CNT = 000002          M.PITM= 000000          NPRT1   004762R     025
F.EFBK= 000010          F.URBD= 000020      L.PRNT= 000001          M.PRDS= 000006          NULL    007364R     025
F.EFN = 000050          F.VAL = 177776      L.PTR = 000000          M.QQAD= 000000          NUMCHN= 000020
F.END = 002000          F.VBN = 000064      L.PTR1= 000004          M.QUAD= 000000          NUMDFL  003230R     025
F.EOBB= 000032          F.VBSZ= 000060      L.SUBS= 000002          M.RAND= 000012          NUMEFL  003233R     025
F.ERR = 000052          GNOME   003102R   C25 MACHIN= 000106           M.RAVL= 000000          NUMESN  003236R     025
F.FACC= 000043          HASHC   005136R   025 MAXCHN= 000014           M.RCIP= 000000          NUMFLG  003230R     025
F.FFBY= 000014          HASHTB  005140R   025 MONBLK  003120R     025 M.RDCT= 000011          NUMLEN  000062R     025
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 45-3
SYMBOL TABLE

NUMNEG  003232R     025 RANDAC  003064R     025 R.LNUM= 000010          SY.BS  = 000010          T.MON  = 000010
NUMQFL  003234R     025 RANDY   003062R     025 R.NEXT= 000006          SY.CLR= 000040          T.NIL  = 000004
NUMRIG  003231R     025 RANERR= 104411          R.OFF  = 000000          SY.ERP= 000020          T.NUM  = 002000
NUMWID  003226R     025 RARGBL  003204R     025 R.PREV= 000004          SY.NOU= 000100          T.OPFN= 000001
NUMZFL  003235R     025 RARGCV  003220R     025 R.PTR  = 000000          SY.OPN= 000002          T.OTHR= 040000
NYIERR= 104432          RARGFL  003202R     025 R.SEQ  = 000003          SY.PMT= 001000          T.SOP  = 000002
N.BLSZ= 000036          RARGLE  003216R     025 R.VAR  = 000002          SY.SRT= 000400          T.SPOP= 000040
N.DESC= 000020          RARGLN  003210R     025 R6     =%000006          SY.SVC= 000004          T.TMP  = 000200
N.DEV  = 000032         RARGPT  003206R     025 R7     =%000007          SY.SVL= 000200          T.VAL  = 100000
N.DEVN= 000034          RARGSF  003214R     025 SI     =%000005          S.AB   = 177776          T.VAR  = 001000
N.DID  = 000024         RARGTY  003212R     025 SIDAMN= 104433           S.ASB  = 100000          UPPER  = 177400
N.DVNM= 000032          RARGVA  003222R     025 SIFRAM  003314R     025 S.BFHD= 000020          UTNAME  000344R     023
N.EXT  = 000014         RBASE   005074R     025 SISTAK  006350R     025 S.BITS= 000002          VALERR= 104413
N.FID  = 000000         RCOUNR  005112R     025 SITOP   006346R     025 S.CHR1= 000003          VARBLE  007400R     025
N.FNAM= 000006          RCOUNT  005110R     025 SITP    007144R     025 S.FATT= 000016          VARBLS  000000R     025
N.FTYP= 000014          REDCNT  005070R     025 SIWARN= 000200           S.FDB  = 000140          VERNUM  000000R     025
N.FVER= 000016          REDIGN  005066R     025 SIZTAB  000044R     025 S.FLB  = 040000          VRSION= 000002
N.NAME= 000006          REDOFF  005072R     025 SI.BIT= 177774           S.FNAM= 000006          WIDTH   000072R     025
N.NEXT= 000022          REENTR  003060R     025 SI.LEN= 177772           S.FNB  = 000036          WROUND  000020R     025
N.PPN  = 000024         RESBIT  003142R     025 SI.RET= 177772           S.FNBW= 000017          WSDATE  000136R     025
N.STAT= 000020          RESCVT  003152R     025 SI.VAL= 177776           S.FNTY= 000004          WSFULL= 104403
N.UNIT= 000034          RESERR= 104440          SKPSIZ  005134R     025 S.FTYP= 000002          WSNAME  000074R     025
OBUFLN= 001000          RESLEN  003150R     025 SPHOLD  000004R     024 S.LEN  = 000002          WSPRT  = ****** GX
OCHAN   005052R     025 RESPTR  003140R     025 SPINIT  000002R     024 S.LINK= 177774          WSTIME  000132R     025
OMODE   000402R     023 RESSFT  003146R     025 STKINC= 000774           S.NFEN= 000020          $FIS   = ****** G
OPTMP0  005066R     025 RESTYP  003144R     025 STOP   = 104420          S.TYPE= 140000          $FPP   = ****** G
OPTMP1  005070R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.VAL  = 000000          $FPPLN= ****** G
OPTMP2  005072R     025 RESVAL  003154R     025 SUBDCT  003106R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP3  005074R     025 RINC    005104R     025 SUBPTR  003110R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP5  005100R     025 ROTINC  005070R     025 SYMF    005340R     025 TMODE   000403R     023 $RT11  = ****** G
OPTMP6  005102R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMPTAG= 000166          .BOOLE= 000002
OPTMP7  005104R     025 ROVER   003312R     025 SYMTAB  005352R     025 TTYBUF  004140R     025 .CHARA= 000000
OPTMP8  005106R     025 RSTS   = 000000          SYMTBF= 104404           TTYLEN= 000622          .DEF.  = 000002
OPTMP9  005110R     025 RSX    = 000115          SYMTOP  005350R     025 TURKEY= 104436          .DLFNB= ****** GX
OPTM10  005112R     025 RSXERR= ****** GX       SYMTP   006272R     025 TYPE   = 000166          .FIND  = ****** GX
OPTM14  005026R     025 RTOP    005070R     025 SYMWRN= 000054           T.BOL  = 020000          .FLOAT= 000006
ORIGIN  000030R     025 RT11   = 000000          SYNERR= 104407           T.CON  = 004000          .FSRCB= ****** G
OUTPTR  000066R     025 R.BITS= 000003           SYSERR= 104402           T.DIM  = 000100          .INTEG= 000004
OVROPR= 000000          R.FIX  = 000001          SYSTAT  000000R     024 T.FUN  = 010000          .MOLUN  000404R     023
PATCH   007344R     025 R.LCT  = 000002          SY.BAD= 000001           T.LIST= 000400          ...GBL= 000000
PRECSN  000052R     025 R.LN   = 000002          SY.BRK= 100000           T.LIT  = 000020          ...TPC= 000020
PRINTM= ****** GX

. ABS.  000000      000
        000000      001
TABLES  000014      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
```

```
DROPC   MACRO M1113  08-FEB-80 11:42  PAGE 45-4
SYMBOL TABLE

TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000160      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12954 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:42
DROPC,DM1:[305,301]DROPC/-SP=APLRX1,APLDEF,DROPC,END
```

M 10

```
**FILE**ID**DYAD
```

```
DDDDDDDD      YY        YY      AAAAAA      DDDDDDDD
DDDDDDDD      YY        YY      AAAAAA      DDDDDDDD
DD       DD   YY        YY    AA      AA    DD       DD
DD       DD   YY        YY    AA      AA    DD       DD
DD       DD     YY    YY      AA      AA    DD       DD
DD       DD     YY    YY      AA      AA    DD       DD
DD       DD       YY          AA      AA    DD       DD
DD       DD       YY          AA      AA    DD       DD
DD       DD       YY          AAAAAAAAAA    DD       DD
DD       DD       YY          AAAAAAAAAA    DD       DD
DD       DD       YY          AA      AA    DD       DD
DD       DD       YY          AA      AA    DD       DD
DDDDDDDD          YY          AA      AA    DDDDDDDD
DDDDDDDD          YY          AA      AA    DDDDDDDD
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 12-FEB-80 07:37
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    DYAD             CALL DYADIC OPERATOR OR FUNCTION
```

DYAD MACRO M1113 12-FEB-80 07:37 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  12-FEB-80 07:37  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                        .TITLE  DYAD
                        .IDENT  \V02.03\

                        .SBTTL  DYAD            CALL DYADIC OPERATOR OR FUNCTION

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;         This routine processes dyadic operator and function calls.
                        ;       For functions and special operators, control is passed to the
                        ;       appropriate routine.  For scalar ops, conformity of the arguments
                        ;       is checked, and the operator routine is called for each element
                        ;       in the arguments.  For scalar ops, the left arg is in AC1, and
                        ;       the right arg is in AC2.  The result is returned  in AC0.  If
                        ;       the operator takes an optional dimension arg, the arg is handled
                        ;       by this routine.  The stack contains (going deeper) the left
                        ;       arg, the operator, optional left bracket followed by dim arg,
                        ;       and finally the right arg.
                        ;
                        ; CALLING SEQUENCE:
                        ;       CALL    DYAD
                        ;
                        ; INPUTS:
                        ;       NONE
                        ;
                        ; IMPLICIT INPUTS:
                        ;       SI      - The SI appears as follows:
                        ;                   (1) Left arg
                        ;                   (2) Operator
                        ;                 Optional:
                        ;                       (3) "["
                        ;                       (4) Dimensional arg
                        ;
                        ;                   (5) Right arg
                        ;
                        ; OUTPUTS:
                        ;       NONE
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;       SI      - The result of the operation appears on the SI.
                        ;
                        ; ROUTINE CALLED:
                        ;       BRKCHK  - Check for CNTL-C interrupt
                        ;       FNCALL  - Used in case of Dyadic function call
                        ;       GETSM1  - Used to get space for the result
                        ;       KILL    - Used to delete unwanted vals on the SI stack
                        ;
                        ; SIDE EFFECTS:
                        ;
                        ; SCRATCH VARABLES USED:
                        ;       OPTMP0  - Left argument SI frame pointer
                        ;       OPTMP1  - Right argument SI frame pointer
                        ;       OPTMP3  - Pointer to argument block which has same size as result
                        ;       OPTMP4  - Pointer to argument block which may be shared with result
                        ;       NONE
                        ;--

000000                  .PSECT CODE
```

```
     59 000000  104407                    DYDSYN: SYNERR                  ;SYNTAX ERROR
     60
     61
     62 000002                              DYAD::  CALL    BRKCHK          ;CHECK TO SEE IF HE WANTS TO STOP
     63 000006                                      PUSH    <R4,R3,R2>      ;SAVE CURRENT TOKEN
     64 000014  005037  003112'                     CLR     DIMBLK          ;CLEAR  DIM ARG BLOCK POINTER
     65 000020  005037  003114'                     CLR     DIMPTR          ;AD VALUE POINTER
     66 000024  052737  100000  004764'            BIS     #F.PRNT,FLAGS   ;PRINT RESULT OF OPERATION
     67 000032  032765  100000  177774            BIT     #T.VAL,SI.BIT(SI);VAL ON SI?
     68 000040  001757                          BEQ     DYDSYN          ;NO, SYNTAX ERROR
     69 000042  016500  177776                  MOV     SI.VAL(SI),R0   ;GET ARG POINTER
     70 000046  010537  005066'                 MOV     SI,OPTMP0       ;SAVE LEFT ARG SI FRAME POINTER
     71 000052  116001  000003                  MOVB    A.BITS(R0),R1   ;GET THE BITS
     72 000056  002164                          BGE     DYDSYS          ;NOT AN ARRAY
     73 000060  042701  177771                  BIC     #^C<A.NUM>,R1   ;CLEAR GARBAGE
     74 000064  010137  003170'                 MOV     R1,LARGTY       ;SAVE THE TYPE
     75 000070  016137  000044' 003172'         MOV     SIZTAB(R1),LARGSF ;LEFT SHIFT CONSTANT
     76 000076  016137  000054' 003174'         MOV     LENTAB(R1),LARGLE ;LEFT LENGTH CONSTANT
     77 000104  016137  000054' 003160'         MOV     LENTAB(R1),LARGFL ;SET ITEM LENGTH
     78 000112  010037  003162'                 MOV     R0,LARGBL       ;SAVE BLOCK POINTER
     79 000116  011037  003164'                 MOV     @R0,LARGPT      ;SAVE VALUE POINTER
     80 000122  016037  000004  003166'         MOV     A.ALEN(R0),LARGLN ;SAVE ARRAY LENGTH
     81 000130  010500                          MOV     SI,R0           ;COPY POINTER
     82 000132  062700  177772                  ADD     #SI.LEN,R0      ;BACK UP 1 FRAME
     83 000136  014004                          MOV     -(R0),R4        ;GET OP NUMBER
     84 000140  014003                          MOV     -(R0),R3        ;GET ITS BITS
     85 000142  032703  000001                  BIT     #T.OPFN,R3      ;IS IT AN OP?
     86 000146  001714                          BEQ     DYDSYN          ;NO, SYNTAX ERROR
     87 000150  032703  000004                  BIT     #T.NIL,R3       ;REALLY A NILADIC FUNCTION?
     88 000154  001311                          BNE     DYDSYN          ;YES, DYADIC USE IS ILLEGAL!!
     89 000156  005740                          TST     -(R0)           ;BACK UP TO FRAME TOP
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 46
DYAD            CALL DYADIC OPERATOR OR FUNCTION

                                        ;NOW THAT WE HAVE POINTERS FOR THE LEFT ARG, CHECK FOR OPTIONAL
                                        ;DIMENSION ARG FOR OPERATOR.
                                        ;
000160  010002                                 MOV     R0,R2           ;SAVE OUR POSITION IN STACK
000162  022740  000073                         CMP     #CHRLBR,-(R0)   ;LEFT BRACKET ON SI?
000166  001033                                 BNE     1$              ;NO, THUS NO DIM ARG
000170  032740  040000                         BIT     #T.OTHR,-(R0)   ;YES, BUT IS IT A CHAR?
000174  001430                                 BEQ     1$              ;NO, MUST BE OP SO NO DIM ARG
000176  005740                                 TST     -(R0)           ;MOVE PAST FRAME
000200  032760  100000  177774                 BIT     #T.VAL,SI.BIT(R0);VAL ON SI?
000206  001504                                 BEQ     4$              ;NO, ERROR
000210  010002                                 MOV     R0,R2           ;SAVE POINTER
000212  014000                                 MOV     -(R0),R0        ;POINT TO DIMENSTION ARG
000214  116001  000003                         MOVB    A.BITS(R0),R1   ;GET THE BITS
000220  002103                                 BGE     DYDSYS          ;IT'S NOT AN ARRAY BLOCK
000222  042701  177771                         BIC     #^C<A.NUM>,R1   ;CLEAR GARBAGE
000226  002477                                 BLT     DYDDOM          ;A CHARACTER Dimension? Rediculous!
000230  010137  003116'                        MOV     R1,DIMTYP       ;STORE THE TYPE
000234  010037  003112'                        MOV     R0,DIMBLK       ;SAVE BLOCK POINTER
000240  011037  003114'                        MOV     @R0,DIMPTR      ;SAVE VALUE POINTER
000244  032703  000100                         BIT     #T.DIM,R3       ;DIM ARG LEGAL?
000250  001463                                 BEQ     4$              ;NO, ERROR AGAIN
000252  062702  177772                         ADD     #SI.LEN,R2      ;BACK POINTER PAST DIM FRAME
                                        ;
                                        ;GET POINTERS FOR RIGHT ARGUMENT
                                        ;
000256  010237  005070'                 1$:     MOV     R2,OPTMP1       ;SAVE RIGHT ARG SI FRAME POINTER
000262  032762  100000  177774                 BIT     #T.VAL,SI.BIT(R2);IS THERE AN ARGUMENT?
000270  001453                                 BEQ     4$              ;NO!
000272  014200                                 MOV     -(R2),R0        ;POINT TO RIGHT ARG
000274  116001  000003                         MOVB    A.BITS(R0),R1   ;GET BITS
000300  002053                                 BGE     DYDSYS          ;NOT ARRAY BLOCK
000302  042701  177771                         BIC     #^C<A.NUM>,R1   ;CLEAR GARBAGE
000306  010137  003212'                        MOV     R1,RARGTY       ;RIGHT ARG TYPE
000312  016137  000044' 003214'                MOV     SIZTAB(R1),RARGSF ;RIGHT SHIFT CONSTANT
000320  016137  000054' 003216'                MOV     LENTAB(R1),RARGLE ;RIGHT LENGTH CONSTANT
000326  016137  000054' 003202'                MOV     LENTAB(R1),RARGFL ;GET LENGTH OF DATA ITEM
000334  010037  003204'                        MOV     R0,RARGBL       ;SAVE BLOCK POINTER
000340  011037  003206'                        MOV     @R0,RARGPT      ;SAVE VALUE POINTER
000344  016037  000004  003210'               MOV     A.ALEN(R0),RARGLN ;SAVE ARRAY LENGTH
                                        ;
                                        ;PROCESS SPECIAL OP OR FUNCTION
                                        ;
000352  032703  000002                         BIT     #T.SOP,R3       ;SCALAR OP?
000356  001025                                 BNE     DYAD1           ;YES, SO WE PROCESS IT
000360  032703  010000                         BIT     #T.FUN,R3       ;NO, FUNCTION?
000364  001407                                 BEQ     3$              ;NO, MUST BE SPECIAL OP
000366  032703  000014                         BIT     #T.NIL+T.MON,R3 ;DYADIC FUNCTION?
000372  001012                                 BNE     4$              ;NO, SYNTAX ERROR
000374                                          CALL    FNCALL          ;CALL DYADIC USER FUNCTION
000400  000137  001162'                 2$:     JMP     DYDEXI          ;AND EXIT
                                        ;
000404                                  3$:     CALL    @DYDTAB(R4)     ;CALL MIXED OR SPECIAL OP
000410  032765  100000  177774                 BIT     #T.VAL,SI.BIT(SI);DID OP ROUTINE  RETURN A VALUE?
000416  001370                                 BNE     2$              ;AND EXIT IF VALUE
000420  104407                          4$:     SYNERR                  ;NO VALUE, SYNTAX ERROR
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 47
DYAD            CALL DYADIC OPERATOR OR FUNCTION

    148 000422  104411                    DYDRAN: RANERR                  ;RANK ERROR
    149 000424  104412                    DYDLEN: LENERR                  ;LENGTH ERROR
    150 000426  104417                    DYDDOM: DOMERR                  ;DOMAIN ERROR
    151
    152 000430  104402                    DYDSYS: SYSERR                  ;ARRRGGGG...
    153                                   ;OP IS SCALAR OP, SO WE MUST CHECK FOR ARGUMENT
    154                                   ;CONFORMITY THEN CALL OP ROUTINE ONCE FOR EACH
    155                                   ;ELEMENT IN THE ARRAY
    156
    157 000432  005037  G05074'           DYAD1:  CLR     OPTMP3          ;POINTER TO ARG TO USE
    158 000436  013702  003162'                   MOV     LARGBL,R2       ;POINT TO LEFT ARG
    159 000442  013703  003204'                   MOV     RARGBL,R3       ;POINT TO RIGHT ARG
    160 000446  022762  000001  000004          CMP     #1,A.ALEN(R2)   ;LEFT ARG SCALAR?
    161 000454  001023                          BNE     2$              ;NO
    162 000456  005037  003160'                 CLR     LARGFL          ;YES, DON'T INCREMENT
    163 000462  022763  000001  000004          CMP     #1,A.ALEN(R3)   ;RIGHT ARG SCALAR?
    164 000470  001007                          BNE     1$              ;NO
    165 000472  005037  003202'                 CLR     RARGFL          ;YES, DON'T INCREMENT
    166 000476  126263  000002  000002          CMPB    A.DIMS(R2),A.DIMS(R3) ;BOTH 1 ELEMENT, WHO MORE DIMS?
    167 000504  003015                          BGT     3$              ;LEFT ARG DOES
    168 000506  001437                          BEQ     6$              ;BOTH THE SAME
    169 000510  010337  005074'           1$:     MOV     R3,OPTMP3       ;POINT TO ''SIMILAR'' BLOCK
    170 000514  013737  005070' 005076'          MOV     OPTMP1,OPTMP4   ;Save pointer to right argument SI frame
    171 000522  000431                          BR      6$
    172
    173 000524  022763  000001  000004  2$:     CMP     #1,A.ALEN(R3)   ;RIGHT ARG SCALAR?
    174 000532  001010                          BNE     4$              ;NO
    175 000534  005037  003202'                 CLR     RARGFL          ;YES, DON'T INCREMENT
    176 000540  010237  005074'           3$:     MOV     R2,OPTMP3       ;POINT TO ''SIMILAR'' BLOCK
    177 000544  013737  005066' 005076'          MOV     OPTMP0,OPTMP4   ;Save pointer to left argument SI frame
    178 000552  000415                          BR      6$
    179
    180 000554  116200  000002            4$:     MOVB    A.DIMS(R2),R0   ;GET # OF DIMS FOR LEFT ARG
    181 000560  120063  000002                    CMPB    R0,A.DIMS(R3)   ;SAME AS RIGHT ARG?
    182 000564  001316                          BNE     DYDRAN          ;RANK ERROR
    183 000566  062702  000006                  ADD     #A.DIM1,R2      ;ADJUST POINTERS
    184 000572  062703  000006                  ADD     #A.DIM1,R3      ;  TO DIM LISTS
    185 000576  022223                    5$:     CMP     (R2)+,(R3)+     ;COMPARE DIM LISTS
    186 000600  001311                          BNE     DYDLEN          ;DIE!
    187 000602                                  SOB     R0,5$           ;LOOP 'TIL DONE
    188
    189 000606  013702  003170'           6$:     MOV     LARGTY,R2       ;GET LEFT TYPE
    190 000612  013703  003212'                   MOV     RARGTY,R3       ;AND RIGHT TYPE
    191 000616  016500  177766                    MOV     SI.BIT+SI.LEN(SI),R0 ;GET BITS OF OP
    192 000622  032700  000020                    BIT     #T.LIT,R0       ;LITERALS ALLOWED?
    193 000626  001004                          BNE     7$              ;YES
    194 000630  005702                          TST     R2              ;LEFT ARG CHAR?
    195 000632  001675                          BEQ     DYDDOM          ;YES, SO DIE
    196 000634  005703                          TST     R3              ;RIGHT ARG SCALAR?
    197 000636  001673                          BEQ     DYDDOM          ;YES, SO DIE
    198 000640  006204                    7$:     ASR     R4              ;POINT TO BYTE, NOT WORD
    199 000642  116400  000000G                 MOVB    DYDBTL(R4),R0   ;SPECIAL STUFF FOR SCALAR OPS
    200 000646  006304                          ASL     R4              ;RE-CONVERT
    201 000650  020203                          CMP     R2,R3           ;Who is Bigger?
    202 000652  002001                          BGE     8$              ;Left is
    203 000654  010302                          MOV     R3,R2           ;No, Right is...
    204 000656                            8$:     CALL    CMPTRT          ;Compute Result Type
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 47-1
DYAD            CALL DYADIC OPERATOR OR FUNCTION

    205 000662  010203                          MOV     R2,R3           ;Copy Intermediate Type
    206 000664  063702  003170'                 ADD     LARGTY,R2       ;Fix Up Left Fetch
    207 000670  063703  003212'                 ADD     RARGTY,R3       ;Fix Up Right Fetch
    208 000674  016237  000000G 003176'         MOV     LARGCT(R2),LARGCV ;FETCH LEFT ROUTINE
    209 000702  016337  000000G 003220'         MOV     RARGCT(R3),RARGCV ;FETCH RIGHT ROUTINE
    210 000710  042701  177771                  BIC     #^C<A.NUM>,R1   ;DE-CONVERT RESULT TYPE
    211 000714  010137  003144'                 MOV     R1,RESTYP       ;STORE RESULT TYPE
    212 000720  013700  005074'                 MOV     OPTMP3,R0       ;IF ONLY 1 ARG IS RESULT ''IMAGE''
    213 000724  001035                          BNE     10$             ;'YOU, YOU'RE THE ONE''
    214
    215 000726  013702  003162'                 MOV     LARGBL,R2       ;POINT TO LEFT ARG
    216 000732  013700  003204'                 MOV     RARGBL,R0       ;  AND RIGHT ARG
    217 000736  013737  005070' 005076'         MOV     OPTMP1,OPTMP4   ;Save pointer to right argument SI frame
    218 000744  132762  000030  000003          BITB    #A.TMP+A.CON,A.BITS(R2) ;IS LEFT ARG TEMPORARY?
    219 000752  001022                          BNE     10$             ;NO, USE RIGHT ARG
    220 000754  132760  000030  000003          BITB    #A.TMP+A.CON,A.BITS(R0) ;IS RIGHT ARG TEMPORARY?
    221 000762  001012                          BNE     9$              ;NO, USE LEFT  ARG
    222 000764  023701  003212'                 CMP     RARGTY,R1       ;SAME TYPE AS RIGHT?
    223 000770  001413                          BEQ     10$             ;YES, USE RIGHT ARG
    224 000772  023701  003170'                 CMP     LARGTY,R1       ;SAME TYPE AS LEFT?
    225 000776  001404                          BEQ     9$              ;YES, USE LEFT ARG
    226 001000  023737  003212' 003170'         CMP     RARGTY,LARGTY   ;OTHERWISE, USE LARGER
    227 001006  002004                          BGE     10$             ;  (I.E., THE RIGHT)
    228 001010  010200                  9$:     MOV     R2,R0           ;USE LEFT ARG
    229 001012  013737  005066' 005076'         MOV     OPTMP0,OPTMP4   ;Save pointer to left argument SI frame
    230
    231 001020                          10$:    PUSH    R0              ;Save model block
    232 001022                                  CALL    GETSM3          ;GET RESULT
    233 001026  026526  177776                  CMP     SI.VAL(SI),(SP)+;Result block same as argument block?
    234 001032  001004                          BNE     16$             ;No, Good
    235 001034  013702  005076'                 MOV     OPTMP4,R2       ;Yes, Get pointer to shared argument SI frame
    236 001040  005062  177774                  CLR     SI.BITS(R2)     ;Clear bits so KILL won't kill array block twice
    237 001044  016037  000004  003224' 16$:    MOV     A.ALEN(R0),DYDCNT ;SAVE COUNT
    238 001052  001443                          BEQ     DYDEXI          ;NOTHING THERE, FINISHED
    239 001054  013702  003164'                 MOV     LARGPT,R2       ;GET POINTER TO LEFT DATA
    240 001060  013703  003206'                 MOV     RARGPT,R3       ;  AND RIGHT DATA
    241
    242 001064                          11$:    CALL    @LARGCV         ;GET 1ST ELEMENT
    243 001070                                  CALL    @RARGCV         ;  FROM EACH
    244 001074  000412                          BR      14$
    245
    246 001076  005737  003160'         12$:    TST     LARGFL          ;ONLY 1?
    247 001102  001402                          BEQ     13$             ;YES
    248 001104                                  CALL    @LARGCV         ;GET NEXT LEFT ELEMENT
    249
    250 001110  005737  003202'         13$:    TST     RARGFL          ;ONLY 1?
    251 001114  001402                          BEQ     14$             ;YES
    252 001116                                  CALL    @RARGCV         ;GET NEXT RIGHT ELEMENT
    253
    254 001122                          14$:    CALL    @DYDTAB(R4)     ;EXECUTE OP
    255 001126                                  CALL    @RESCVT         ;STORE RESULT
    256 001132  000404                          BR      15$             ;ERROR, OVERFLOW
    257 001134  005337  003224'                 DEC     DYDCNT          ;DECREMENT COUNT
    258 001140  001356                          BNE     12$             ;'TILL DONE
    259 001142  000407                          BR      DYDEXI          ;..DONE..
    260                                 ;
    261                                 ;THIS CODE HANDLES OVERFLOWS, SUCH AS ADDING TWO LARGE INTEGERS
```

```
                                        ;TOGETHER AND GETTING A VALUE LARGER THAN 2**15-1
                                        ;
001144                          15$:    CALL    DYDEXP          ;EXPAND INTEGER TO FLOAT
001150  163702  003160'                 SUB     LARGFL,R2       ;BACK UP ONE
001154  163703  003202'                 SUB     RARGFL,R3       ;  FOR SOURCES
001160  000741                         BR      11$             ;AND CONTINUE
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 48
DYAD            CALL DYADIC OPERATOR OR FUNCTION

                                        ;POP ARGS FROM STACK, RELEASE THEM, PROMOTE
                                        ;RESULT AND RETURN

001162  014537  003136'        DYDEXI: MOV     -(SI),RESULT    ;POINT TO RESULT
001166  014537  003142'                MOV     -(SI),RESBIT    ;GET BITS
001172  005745                         TST     -(SI)           ;SKIP OVER RETURN
001174  013700  003136'                MOV     RESULT,R0       ;POINT TO THE RESULT BLOCK
001200  152760  000010  000003        BISB    #A.TMP,A.BITS(R0);PROTECT THE RESULT BLOCK FROM BEING KILLED
001206                                 CALL    KILL            ;KILL LEFT ARG
001212                                 CALL    KILL            ;KILL OP
001216  005737  003112'                TST     DIMBLK          ;DIM ARG?
001222  001404                         BEQ     1$              ;NO
001224                                 CALL    KILL            ;KILL BRACKET
001230                                 CALL    KILL            ;KILL DIM ARG
001234                          1$:    CALL    KILL            ;KILL RIGHT ARG
001240  005725                         TST     (SI)+           ;SKIP OVER RETURN
001242  013725  003142'                MOV     RESBIT,(SI)+    ;SET RESULT BITS
001246  013700  003136'                MOV     RESULT,R0       ;POINT TO RESULT
001252  142760  000010  000003        BICB    #A.TMP,A.BITS(R0);RESULT IS ALWAYS TEMP
001260  010025                         MOV     R0,(SI)+        ;POINT TO RESULT
001262                                 POP     <R2,R3,R4>      ;RESTORE TOKEN
001270  005037  003112'                CLR     DIMBLK          ;PREVENT PROBLEMS
001274                                 RETURN
```

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 50
DYAD            CALL DYADIC OPERATOR OR FUNCTION

      1         000001                          .END
```

DYAD MACRO M1113 12-FEB-80 07:37 PAGE 50-1
SYMBOL TABLE

```
AC0    =%000000          CHRFRO= 000135    C.ALOC= 000162    D.BXCT= 000007    D.$MIN= 000000
AC1    =%000001          CHRLBR= 000073    C.BLOK= 000064    D.BXIO= 000007    D.$MUL= 000000
AC2    =%000002          CHRLPR= 000072    C.BLSZ= 000140    D.BXNL= 000007    D.$PLU= 000000
AC3    =%000003          CHRMNS= 000137    C.BUFF= 177760    D.BXPP= 000007    D.$RES= 000000
AC4    =%000004          CHRMUL= 000075    C.BUFL= 177762    D.BXPW= 000007    D.$SUB= 000000
AC5    =%000005          CHRNEG= 000100    C.CCNT= 177766    D.BXRL= 000007    D.1COM= 000001
ALLINT  003274R    025   CHRPDL= 000271    C.CPTR= 177764    D.CIRC= 000001    D.1EXP= 000002
ARYRET  003276R    025   CHRPLU= 000055    C.CZ  = 001000    D.COMB= 000001    D.1ROT= 000003
ASCQUO= 000042           CHRPWR= 000120    C.DIG = 000040    D.COMP= 000001    EDICHR  005063R    025
A.ALEN= 000004           CHRQST= 000121    C.FASW= 020000    D.CONT= 000006    EDICNT  005064R    025
A.ARG = 000040           CHRQUO= 000113    C.FLEN= 000010    D.CTLM= 000000    EDIFLG  005062R    025
A.ASB = 000200           CHRRBR= 000047    C.FLSH= 004000    D.DEAL= 000003    EDIT   = 000003
A.BITS= 000003           CHRRPR= 000042    C.FRC = 000001    D.DECD= 000002    EIS    = 000000
A.CON = 000020           CHRSEM= 000074    C.IDEN= 000010    D.DIV = 000000    ELESIZ  005132R    025
A.DIMS= 000002           CHRSMO= 000112    C.NAME= 000102    D.DMD = 000010    ENDCOR  000004R    025
A.DIM1= 000006           CHRSPA= 000040    C.NOBF= 000400    D.DQAD= 177777    EOF    = 104475
A.NUM = 000006           CHRUPA= 000131    C.NUM = 000020    D.DROP= 000002    ERRMAX= 000166
A.PTR = 000000           CHR.BS= 000010    C.OP  = 000004    D.ENCD= 000002    ERRNUM  004772R    025
A.TMP = 000010           CHR.CC= 000003    C.OPN = 010000    D.EQ  = 000000    ERRPTR  004770R    025
BADDIR= 104463           CHR.CR= 000015    C.RDLN= 177776    D.EXPO= 000002    ERR0    004774R    025
BADERR= 104437           CHR.CZ= 000032    C.REWT= 002000    D.FGET= 000005    EXEBUF  005054R    025
BLZBUB  005116R    025   CHR.GT= 000046    C.RND = 000010    D.FMT = 000004    E$CT    005056R    025
BRKCHK= ****** GX        CHR.LF= 000012    C.SBIT= 177756    D.FPNT= 000005    E$PT    005060R    025
BRSELF  005046R    025   CHR.LT= 000043    C.SPFX= 000100    D.FPUT= 000005    FA.APD= 000100
CCNT    005072R    025   CHR.SI= 000017    C.SPGL= 000200    D.GE  = 000000    FA.CRE= 000010
CHANEX  003060R    025   CHR.SL= 000057    C.SPLX= 000001    D.GT  = 000000    FA.DLK= 001000
CHAN00  000256R    025   CHR.SO= 000016    C.SPRS= 000002    D.IBM = 000002    FA.ENB= 100000
CHAN01  000330R    025   CHR.TB= 000011    C.SRC = 177774    D.IND = 000002    FA.EXC= 002000
CHAN02  000512R    025   CHR.UL= 000106    C.SRCP= 000024    D.INNR= 000013    FA.EXT= 000004
CHAN03  000674R    025   CHR..A= 000141    C.TT  = 000004    D.INTR= 000006    FA.NSP= 000100
CHAN04  001056R    025   CHR..E= 000145    C.WIDC= 177770    D.LE  = 000000    FA.NSR= 000200
CHAN05  001240R    025   CHR..S= 000163    C.WIDR= 177772    D.LOG = 000001    FA.POS= 010000
CHAN06  001422R    025   CHR..T= 000164    DEBUG = 000000    D.LT  = 000000    FA.RD = 000001
CHAN07  001604R    025   CHR..U= 000165    DEFERR= 104405    D.MAX = 000000    FA.RWD= 004000
CHAN08  001766R    025   CHR..Z= 000172    DEVIL   005114R    025 D.MEM = 000002    FA.SEQ= 040000
CHAN09  002150R    025   CHR..0= 000060    DEVNTF= 104470    D.MIN = 000000    FA.SHR= 000040
CHAN10  002332R    025   CHR..1= 000061    DIGITS  000006R    025 D.MUL = 000000    FA.TMP= 000020
CHAN11  002514R    025   CH.AND= 000001    DIMAX = 000020    D.NE  = 000000    FA.WCK= 020000
CHAN12  002676R    025   CMDBUF  003250R    025 DIMBLK  003112R    025 D.NN  = 000000    FA.WRT= 000002
CHAN13  000022R    023   CMDMAX= 000010    DIMPTR  003114R    025 D.NR  = 000000    FD.BLK= 000010
CHAN14  000204R    023   CMPTRT= ****** GX DIMTYP  003116R    025 D.OR  = 000000    FD.CCL= 000002
CHNOPN= 104471           CNTSIZ  005130R    025 DIRFUL= 104466    D.PLUS= 000000    FD.COM= 020000
CHRAND= 000051           COMBIN  005076R    025 DOMERR= 104417    D.PSUB= 000006    FD.CR = 000002
CHRBEL= 000007           COMCNT  005126R    025 DPTERR= 104416    D.PWR = 000001    FD.DIR= 000010
CHRBKR= 000133           COMERR= 104425    DYAD    000002RG    020 D.QQAD= 000000    FD.FTN= 000001
CHRBOL= 000200           COMPTR  005122R    025 DYAD1   000432R    020 D.QUAD= 000000    FD.F11= 040000
CHRBOX= 000114           COMRTN  003156R    025 DYDBTL= ****** GX    D.RES = 000000    FD.INS= 000010
CHRCMA= 000054           COMSKP  005124R    025 DYDCNT  003224R    025 D.RHO = 000000    FD.ISP= 002000
CHRCOL= 000076           CORPTR  005120R    025 DYDDOM  000426R    020 D.ROT = 000003    FD.MNT= 100000
CHRDEL= 000107           CS.ASS= 000200    DYDEXI  001162R    020 D.SCON= 000006    FD.OSP= 004000
CHRDIV= 000053           CS.BOO= 000010    DYDEXP= ****** GX    D.SDIF= 000006    FD.PLC= 000004
CHRDLR= 000176           CS.FLO= 000020    DYDLEN  000424R    020 D.SUB = 000000    FD.PRN= 000004
CHRDLT= 000110           CS.IDN= 000040    DYDRAN  000422R    020 D.SUBS= 000006    FD.PSE= 010000
CHRDOT= 000056           CS.ONE= 000100    DYDSYN  000000R    020 D.TAKE= 000002    FD.RAH= 000001
CHREP = 000105           CTEPS   000152R    025 DYDSYS  000430R    020 D.TRAN= 000004    FD.RAN= 000002
CHRERR= 104431           CTMAX   000212R    025 DYDTAB= ****** GX    D.UNON= 000006    FD.REC= 000001
CHRFIL= 000177           CTRMAX  000172R    025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
```

SYMBOL TABLE

```
FD.SDI= 000020          F.DFNB= 000046          F.SUSP= 000002          LOCLCT  005036R  025 M.NEG = 000000
FD.SQD= 000040          F.DSPT= 000044          F.TRC = 000002          LOWCOR  000002R  025 M.NOT = 000000
FD.TTY= 000004          F.DVNM= 000134          F.UNIT= 000136          LSTERP  004766R  025 M.NUB = 000006
FD.WBH= 000002          F.EFBK= 000010          F.URBD= 000020          LWIDTH  000070R  025 M.NUM = 000004
FF.CHR= 000005          F.EFN = 000050          F.VAL = 177776          L.BITS= 000003      M.OUTR= 000012
FF.NV = 000003          F.END = 002000          F.VBN = 000064          L.CNT = 000002      M.PITM= 000000
FF.POE= 000002          F.EOBB= 000032          F.VBSZ= 000060          L.PRNT= 000001      M.PRDS= 000006
FF.RWD= 000001          F.ERR = 000052          GETSM3= ****** GX       L.PTR = 000000      M.QQAD= 000000
FF.RWF= 000006          F.FACC= 000043          GNOME   003102R  025    L.PTR1= 000004      M.QUAD= 000000
FF.SPC= 000004          F.FFBY= 000014          HASHC   005136R  025    L.SUBS= 000002      M.RAND= 000012
FHEADR  005012R  025    F.FLAG= 000002          HASHTB  005140R  025    MACHIN= 000106      M.RAVL= 000000
FHRET   005040R  025    F.FLB = 000100          HSHMSK= 177601          MAXCHN= 000014      M.RCIP= 000000
FILNTF= 104467          F.FNAM= 000110          HSHSIZ= 000200          MONBLK  003120R  025 M.RDCT= 000011
FIS   = 000000          F.FNB = 000102          ICHAN   005050R  025    MONCVT  003132R  025 M.REVS= 000003
FISCON= 046000          F.FNDF= 000010          ILFNAM= 104464          MONLEN  003130R  025 M.ROLL= 000000
FIXPTR  003074R  025    F.FTYP= 000116          ILLCHN= 104540          MONPTR  003122R  025 M.SATA= 000001
FLAGS   004764R  025    F.FUN = 000010          ILLSW = 104565          MONSFT  003126R  025 M.SCAN= 000011
FLCBUF  003250R  025    F.FVER= 000120          INDERR= 104410          MONTYP  003124R  025 M.SHAP= 000000
FLINE   005014R  025    F.HEAD= 000006          INNTMP  003250R  025    MONVAL  003134R  025 M.SIGN= 000000
FLOEXP  003270R  025    F.HIBK= 000004          INNTM2  005016R  025    M.ABS = 000000      M.SUBR= 000001
FLONEG  003272R  025    F.LBL = 000020          INPTR   000064R  025    M.BOL = 000200      M.SYM = 000203
FLSWTB  000222R  025    F.LINE= 000012          INSNAM= 104462          M.BXAV= 000007      M.SYN = 000201
FLTLEN= 000010          F.LOC = 000004          IOERR = 104477          M.BXCR= 000007      M.TRAN= 000012
FNCALL= ****** GX       F.LSTC= 000004          I.CNT = 177776          M.BXCT= 000007      M.$ABS= 000000
FNINSI  005044R  025    F.LUN = 000042          I.DIM = 177772          M.BXEX= 000007      M.$CEI= 000000
FORIGN  000032R  025    F.MBCT= 000054          I.LEN = 000012          M.BXFX= 000007      M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062          M.BXIO= 000007      M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770          M.BXLC= 000003      M.$NEG= 000000
FO.RD = 000001          F.MON = 000002          I.PTR = 177774          M.BXNC= 000007      M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766          M.BXNL= 000007      M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024          KILL  = ****** GX       M.BXPP= 000007      M.1REV= 000003
FPMODE  000042R  025    F.NREC= 000030          LABERR= 104406          M.BXPW= 000007      M.1SCN= 000011
FPP   = 000000          F.OPN = 000010          LABLPT  005042R  025    M.BXRL= 000007      NAMBUF  003240R  025
FPPLEN= 000004          F.OVBS= 000030          LARGBL  003162R  025    M.BXTS= 000007      NB.DEV= 000200
FREBLK  007400R  025    F.PREV= 000000          LARGCT= ****** GX       M.BXWA= 000007      NB.DIR= 000100
FUZZ    000142R  025    F.PRNT= 100000          LARGCV  003176R  025    M.CEI = 177777      NB.NAM= 000004
FXPTR1  003076R  025    F.PROT= 010000          LARGFL  003160R  025    M.CHR = 000202      NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016          LARGLE  003174R  025    M.CMT = 000206      NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001          LARGLN  003166R  025    M.DOUT= 000006      NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034          LARGPT  003164R  025    M.DQAD= 000006      NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017          LARGSF  003172R  025    M.ENCD= 000004      NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000          LARGTY  003170R  025    M.EXE = 000000      NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040          LARGVA  003200R  025    M.EXP = 000001      NB.VER= 000001
F.BBFS= 000062          F.RES = 000040          LBINC   005100R  025    M.FACT= 000001      ND.DEV= 000200
F.BDB = 000070          F.RET = 177772          LBUFF   003316R  025    M.FIL = 000177      ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002          LBUFLN= 000621          M.FLR = 177777      ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000          LCOUNT  005106R  025    M.FMT = 000004      ND.PPN= 000100
F.BKDN= 000026          F.R2  = 000014          LENERR= 104412          M.GDUP= 000003      ND.$EX= 000000
F.BKDS= 000020          F.R3  = 000016          LENTAB  000054R  025    M.GDWN= 000003      ND.$ES= 000020
F.BKEF= 000050          F.R4  = 000020          LEXPTR  003100R  025    M.GETI= 000006      ND.$NM= 000000
F.BKP1= 000051          F.SEQN= 000100          LFTOP   005066R  025    M.IBM = 000003      ND.$NS= 000040
F.BKST= 000024          F.SPDV= 000072          LINC    005102R  025    M.IDN = 000000      ND.$PN= 001400
F.BKVB= 000064          F.SPUN= 000074          LINE    005016R  025    M.INSP= 000006      NEGSGN  005063R  025
F.BR  = 000020          F.STBK= 000036          LININC  005026R  025    M.IOTA= 000000      NINPT1  003302R  025
F.CHR = 000075          F.STK = 000022          LISIZE= 000020          M.LIT = 000205      NINPT2  003300R  025
F.CNTG= 000034          F.STOP= 000004          LITOP   005072R  025    M.LN  = 000001      NINSTK  003304R  025
F.CODE= 000010          F.STP = 000001          LNLNG = 104541          M.MMD = 000010      NINTMP  003310R  025
```

DYAD MACRO M1113 12-FEB-80 07:37 PAGE 50-3
SYMBOL TABLE

```
NINTYP  003306R      025 OPTMP8  005106R      025 RSX   = 000115          SYNERR= 104407          T.DIM = 000100
NOTAVL= 104472           OPTMP9  005110R      025 RTOP    005070R   025 SYSERR= 104402          T.FUN = 010000
NOTOPN= 104473           OPTM10  005112R      025 RT11  = 000000          SYSTAT  000000R   024 T.LIST= 000400
NPRT1   004762R      025 OPTM14  005026R      025 R.BITS= 000003          SY.BAD= 000001          T.LIT = 000020
NULL    007364R      025 ORIGIN  000030R      025 R.FIX = 000001          SY.BRK= 100000          T.MON = 000010
NUMCHN= 000020           OUTPTR  000066R      025 R.LCT = 000002          SY.BS = 000010          T.NIL = 000004
NUMDFL  003230R      025 OVROPR= 000000           R.LN  = 000002          SY.CLR= 000040          T.NUM = 002000
NUMEFL  003233R      025 PATCH   007344R      025 R.LNUM= 000010          SY.ERP= 000020          T.OPFN= 000001
NUMESN  003236R      025 PRECSN  000052R      025 R.NEXT= 000006          SY.NOU= 000100          T.OTHR= 040000
NUMFLG  003230R      025 RANDAC  003064R      025 R.OFF = 000000          SY.OPN= 000002          T.SOP = 000002
NUMLEN  000062R      025 RANDY   003062R      025 R.PREV= 000004          SY.PMT= 001000          T.SPOP= 000040
NUMNEG  003232R      025 RANERR= 104411           R.PTR = 000000          SY.SRT= 000400          T.TMP = 000200
NUMQFL  003234R      025 RARGBL  003204R      025 R.SEQ = 000003          SY.SVC= 000004          T.VAL = 100000
NUMRIG  003231R      025 RARGCT= ****** GX        R.VAR = 000002          SY.SVL= 000200          T.VAR = 001000
NUMWID  003226R      025 RARGCV  003220R      025 R6    =%000006          S.AB  = 177776          UPPER = 177400
NUMZFL  003235R      025 RARGFL  003202R      025 R7    =%000007          S.ASB = 100000          UTNAME  000344R   023
NYIERR= 104432           RARGLE  003216R      025 SI    =%000005          S.BFHD= 000020          VALERR= 104413
N.BLSZ= 000036           RARGLN  003210R      025 SIDAMN= 104433          S.BITS= 000002          VARBLE  007400R   025
N.DESC= 000020           RARGPT  003206R      025 SIFRAM  003314R   025 S.CHR1= 000003          VARBLS  000000R   025
N.DEV = 000032           RARGSF  003214R      025 SISTAK  006350R   025 S.FATT= 000016          VERNUM  000000R   025
N.DEVN= 000034           RARGTY  003212R      025 SITOP   006346R   025 S.FDB = 000140          VRSION= 000002
N.DID = 000024           RARGVA  003222R      025 SITP    007144R   025 S.FLB = 040000          WIDTH   000072R   025
N.DVNM= 000032           RBASE   005074R      025 SIWARN= 000200          S.FNAM= 000006          WROUND  000020R   025
N.EXT = 000014           RCOUNR  005112R      025 SIZTAB  000044R   025 S.FNB = 000036          WSDATE  000136R   025
N.FID = 000000           RCOUNT  005110R      025 SI.BIT= 177774          S.FNBW= 000017          WSFULL= 104403
N.FNAM= 000006           REDCNT  005070R      025 SI.LEN= 177772          S.FNTY= 000004          WSNAME  000074R   025
N.FTYP= 000014           REDIGN  005066R      025 SI.RET= 177772          S.FTYP= 000002          WSTIME  000132R   025
N.FVER= 000016           REDOFF  005072R      025 SI.VAL= 177776          S.LEN = 000002          $FIS  = ****** G
N.NAME= 000006           REENTR  003060R      025 SKPSIZ  005134R   025 S.LINK= 177774          $FPP  = ****** G
N.NEXT= 000022           RESBIT  003142R      025 SPHOLD  000004R   024 S.NFEN= 000020          $FPPLN= ****** G
N.PPN = 000024           RESCVT  003152R      025 SPINIT  000002R   024 S.TYPE= 140000          $MACHN= ****** G
N.STAT= 000020           RESERR= 104440               STKINC= 000774          S.VAL = 000000          $RSTS = ****** G
N.UNIT= 000034           RESLEN  003150R      025 STOP  = 104420          TKDIMS  005072R   025 $RSX  = ****** G
OBUFLN= 001000           RESPTR  003140R      025 SUBCNT  003104R   025 TKFRAM  005066R   025 $RT11 = ****** G
OCHAN   005052R      025 RESSFT  003146R      025 SUBDCT  003106R   025 TKLEFT  005070R   025 .BOOLE= 000002
OMODE   000402R      023 RESTYP  003144R      025 SUBPTR  003110R   025 TMODE   000403R   023 .CHARA= 000000
OPTMP0  005066R      025 RESULT  003136R      025 SYMB1   005344R   025 TMPTAG= 000166           .DEF. = 000002
OPTMP1  005070R      025 RESVAL  003154R      025 SYMF    005340R   025 TTYBUF  004140R   025 .FLOAT= 000006
OPTMP2  005072R      025 RINC    005104R      025 SYMPT1  005346R   025 TTYLEN= 000622           .FSRCB= ****** G
OPTMP3  005074R      025 RLTBAS  005066R      025 SYMTAB  005352R   025 TURKEY= 104436           .INTEG= 000004
OPTMP4  005076R      025 ROTINC  005070R      025 SYMTBF= 104404          TYPE  = 000166          .MOLUN  000404R   023
OPTMP5  005100R      025 ROUND   000010R      025 SYMTOP  005350R   025 T.BOL = 020000          ...GBL= 000000
OPTMP6  005102R      025 ROVER   003312R      025 SYMTP   006272R   025 T.CON = 004000          ...TPC= 000020
OPTMP7  005104R      025 RSTS  = 000000           SYMWRN= 000054

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
```

D 12

```
DYAD    MACRO M1113  12-FEB-80 07:37  PAGE 50-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    001276      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13088 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:58
DYAD,DM1:[305,301]DYAD/-SP=APLRX1,APLDEF,DYAD,END
```

E 12

E 12

FILEID**DYDEXP

```
DDDDDDDD      YY       YY   DDDDDDDD      EEEEEEEEEE   XX       XX   PPPPPPPP
DDDDDDDD      YY       YY   DDDDDDDD      EEEEEEEEEE   XX       XX   PPPPPPPP
DD       DD   YY       YY   DD       DD   EE           XX       XX   PP       PP
DD       DD   YY       YY   DD       DD   EE           XX       XX   PP       PP
DD       DD     YY   YY     DD       DD   EE             XX   XX     PP       PP
DD       DD     YY   YY     DD       DD   EE             XX   XX     PP       PP
DD       DD       YY       DD       DD   EEEEEEEE         XX       PPPPPPPP
DD       DD       YY       DD       DD   EEEEEEEE         XX       PPPPPPPP
DD       DD       YY       DD       DD   EE             XX   XX     PP
DD       DD       YY       DD       DD   EE             XX   XX     PP
DD       DD       YY       DD       DD   EE           XX       XX   PP                  ....
DD       DD       YY       DD       DD   EE           XX       XX   PP                  ....
DDDDDDDD          YY       DDDDDDDD      EEEEEEEEEE   XX       XX   PP                  ....
DDDDDDDD          YY       DDDDDDDD      EEEEEEEEEE   XX       XX   PP                  ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

F 12

.MAIN. MACRO M1113 08-FEB-80 11:46
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    DYDEXP           EXPAND INTEGER RESULT TO FLOAT
```


DYDEXP MACRO M1113 08-FEB-80 11:46 PAGE 1

DYDEXP MACRO M1113 08-FEB-80 11:46 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

DYDEXP MACRO M1113 08-FEB-80 11:46 PAGE 2
APLDEF

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:46  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
DYDEXP  MACRO M1113  08-FEB-80 11:46  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  DYDEXP
                                .IDENT  \V02.03\

                                .SBTTL  DYDEXP          EXPAND INTEGER RESULT TO FLOAT

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;         This Routine is Called from within a Dyadic Execution
                                ;       loop. Hence, R0 is unknown data, R1 a Pointer to the
                                ;       Left Arg, R2 a pointer to the Right arg, R3 a Pointer
                                ;       to the Result, and R4 the Op Number (+/-). This Routine
                                ;       allocates a new block, converts from Integer to Float,
                                ;       deletes the old block, gets pointers to the new routines,
                                ;       and corrects the registers. fun, huh?
                                ;
                                ; INPUTS:
                                ;       R1      - Pointer to Left Arg
                                ;       R2      - Pointer to Right Arg
                                ;       R3      - Pointer to Result
                                ;       R4      - Op Number
                                ;
                                ; OUTPUTS:
                                ;       R1      - Pointer to Left Arg (back up once)
                                ;       R2      - Pointer to Right Arg (back up once)
                                ;       R3      - Pointer to Result (new data area)
                                ;       R4      - Op Number (probably negated)
                                ;--

000000                          .PSECT  CODE

000000                          DYDEXP::
000000  023727  003144' 000004          CMP     RESTYP,#.INTEGER ;source integer?
000006  001102                        BNE     6$              ;no, system error - die!
000010                          PUSH    <R0,R3,R2,R1>   ;save regs
000020  016500  177776          MOV     SI.VAL(SI),R0   ;get pointer to result
000024  012701  000006          MOV     #.FLOAT,R1      ;result type
000030  010137  003144'         MOV     R1,RESTYP       ;save new result type
000034  016137  000000G 003152' MOV     FRESCT(R1),RESCVT ;new store routine
000042  013702  003170'         MOV     LARGTY,R2       ;get left arg type
000046  016237  000000G 003176' MOV     FLARGC(R2),LARGCV ;new left fetch routine
000054  013702  003212'         MOV     RARGTY,R2       ;get right arg type
000060  016237  000000G 003220' MOV     FRARGC(R2),RARGCV ;new right fetch routine
000066  005704                  TST     R4              ;negative?
000070  002404                  BLT     1$              ;yes, don't change it
000072  005404                  NEG     R4              ;make it negative
000074  005764  000000G         TST     DYDTAB(R4)      ;any routine?
000100  001445                  BEQ     6$              ;nope, die
000102  011002          1$:     MOV     (R0),R2         ;save pointer to old data
000104                          CALL    GETSM2          ;get new result block
000110                          POP     R3              ;ending pointer
000112  020203          2$:     CMP     R2,R3           ;finished?
000114  001407                  BEQ     3$              ;yes
000116                          LDCIF   (R2)+,AC3       ;get integer
000124                          STF     AC3,(R1)+       ;store float
000132  000767                  BR      2$              ;loop
000134                  3$:     PUSH    R1              ;save new r1
000136                          PUSH    <-(SI),-(SI),-(SI)> ;save new result block
```

K 12

```
DYDEXP  MACRO M1113  08-FEB-80 11:46  PAGE 44-1
DYDEXP           EXPAND INTEGER RESULT TO FLOAT

     58 000144  026537  177776  003162'          CMP     SI.VAL(SI),LARGBL ;same as left arg?
     59 000152  001404                           BEQ     4$                ;yes, don't kill
     60 000154  026537  177776  003204'          CMP     SI.VAL(SI),RARGBL ;same as right arg?
     61 000162  001002                           BNE     5$                ;no, kill it
     62 000164  005065  177774            4$:    CLR     SI.BIT(SI)        ;screw around
     63 000170                            5$:    CALL    KILL              ;die, die, die
     64 000174                                   POP     <(SI)+,(SI)+,(SI)+,R1,R2,R3,R0> ;restore stuff
     65 000212                                   RETURN                    ;go, babe
     66
     67 000214  104402                    6$:    SYSERR                    ;system error, die
```

L 12

L 12

```
DYDEXP  MACRO M1113  08-FEB-80 11:46  PAGE 46
DYDEXP          EXPAND INTEGER RESULT TO FLOAT

      1         000001                          .END
```

DYDEXP MACRO M1113 08-FEB-80 11:46 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DEAL= 000003    EDIT  = 000003
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DECD= 000002    EIS   = 000000
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DIV = 000000    ELESIZ  005132R     025
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DMD = 000010    ENDCOR  000004R     025
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.DQAD= 177777    EOF   = 104475
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.DROP= 000002    ERRMAX= 000166
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.ENCD= 000002    ERRNUM  004772R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.EQ  = 000000    ERRPTR  004770R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.EXPD= 000002    ERRO    004774R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FGET= 000005    EXEBUF  005054R     025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FMT = 000004    E$CT    005056R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.FPNT= 000005    E$PT    005060R     025
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.FPUT= 000005    FA.APD= 000100
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.GE  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.GT  = 000000    FA.DLK= 001000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.IBM = 000002    FA.ENB= 100000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.IND = 000002    FA.EXC= 002000
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.INNR= 000013    FA.EXT= 000004
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.INTR= 000006    FA.NSP= 000100
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LE  = 000000    FA.NSR= 000200
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.LOG = 000001    FA.POS= 010000
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.LT  = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MAX = 000000    FA.RWD= 004000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MEM = 000002    FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.MIN = 000000    FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.MUL = 000000    FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NE  = 000000    FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.NN  = 000000    FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.NR  = 000000    FD.BLK= 000010
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.OR  = 000000    FD.CCL= 000002
CHAN05  001240R     025  CHR..S= 000163    DEBUG = 000000    D.PLUS= 000000    FD.COM= 020000
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.PSUB= 000006    FD.CR = 000002
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R     025 D.PWR = 000001    FD.DIR= 000010
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.QQAD= 000000    FD.FTN= 000001
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R     025 D.QUAD= 000000    FD.F11= 040000
CHAN10  002332R     025  CHR..1= 000061    DIMAX = 000020    D.RES = 000000    FD.INS= 000010
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R     025 D.RHO = 000000    FD.ISP= 002000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.ROT = 000003    FD.MNT= 100000
CHAN13  000022R     023  CMDMAX= 000010    DIMTYP  003116R     025 D.SCON= 000006    FD.OSP= 004000
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466    D.SDIF= 000006    FD.PLC= 000004
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417    D.SUB = 000000    FD.PRN= 000004
CHRAND= 000051           COMCNT  005126R     025 DPTERR= 104416    D.SUBS= 000006    FD.PSE= 010000
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R     025 D.TAKE= 000002    FD.RAH= 000001
CHRBKR= 000133           COMPTR  005122R     025 DYDEXP  000000RG    020 D.TRAN= 000004    FD.RAN= 000002
CHRBOL= 000200           COMRTN  003156R     025 DYDTAB= ****** GX     D.UNON= 000006    FD.REC= 000001
CHRBOX= 000114           COMSKP  005124R     025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRCMA= 000054           CORPTR  005120R     025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRCOL= 000076           CS.ASS= 000200        D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRDEL= 000107           CS.BOO= 000010        D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRDIV= 000053           CS.FLO= 000020        D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDLR= 000176           CS.IDN= 000040        D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDLT= 000110           CS.ONE= 000100        D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDOT= 000056           CTEPS   000152R     025 D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHREP = 000105           CTMAX   000212R     025 D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRERR= 104431           CTRMAX  000172R     025 D.COMP= 000001    EDICHR  005063R     025 FF.RWF= 000006
CHRFIL= 000177           C.ALOC= 000162        D.CONT= 000006    EDICNT  005064R     025 FF.SPC= 000004
CHRFRO= 000135           C.BLOK= 000064        D.CTLM= 000000    EDIFLG  005062R     025 FHEADR  005012R     025
```

SYMBOL TABLE

```
FHRET   005040R     025 F.FFBY= 000014        HASHC   005136R     025 MACHIN= 000106        M.RAVL= 000000
FILNTF= 104467          F.FLAG= 000002        HASHTB  005140R     025 MAXCHN= 000014        M.RCIP= 000000
FIS   = 000000          F.FLB = 000100        HSHMSK= 177601          MONBLK  003120R     025 M.RDCT= 000011
FISCON= 046000          F.FNAM= 000110        HSHSIZ= 000200          MONCVT  003132R     025 M.REVS= 000003
FIXPTR  003074R     025 F.FNB = 000102        ICHAN   005050R     025 MONLEN  003130R     025 M.ROLL= 000000
FLAGS   004764R     025 F.FNDF= 000010        ILFNAM= 104464          MONPTR  003122R     025 M.SATA= 000001
FLARGC= ****** GX       F.FTYP= 000116        ILLCHN= 104540          MONSFT  003126R     025 M.SCAN= 000011
FLCBUF  003250R     025 F.FUN = 000010        ILLSW = 104565          MONTYP  003124R     025 M.SHAP= 000000
FLINE   005014R     025 F.FVER= 000120        INDERR= 104410          MONVAL  003134R     025 M.SIGN= 000000
FLOEXP  003270R     025 F.HEAD= 000006        INNTMP  003250R     025 M.ABS = 000000        M.SUBR= 000001
FLONEG  003272R     025 F.HIBK= 000004        INNTM2  005016R     025 M.BOL = 000200        M.SYM = 000203
FLSWTB  000222R     025 F.LBL = 000020        INPTR   000064R     025 M.BXAV= 000007        M.SYN = 000201
FLTLEN= 000010          F.LINE= 000012        INSNAM= 104462          M.BXCR= 000007        M.TRAN= 000012
FNINSI  005044R     025 F.LOC = 000004        IOERR = 104477          M.BXCT= 000007        M.$ABS= 000000
FORIGN  000032R     025 F.LSTC= 000004        I.CNT = 177776          M.BXEX= 000007        M.$CEI= 000000
FO.APD= 000100          F.LUN = 000042        I.DIM = 177772          M.BXFX= 000007        M.$FLR= 000000
FO.MFY= 000002          F.MBCT= 000054        I.LEN = 000012          M.BXIO= 000007        M.$IDN= 000000
FO.RD = 000001          F.MBC1= 000055        I.OERR= 000062          M.BXLC= 000003        M.$NEG= 000000
FO.UPD= 000006          F.MBFG= 000056        I.OLD = 177770          M.BXNC= 000007        M.$SIG= 000000
FO.WRT= 000016          F.MON = 000002        I.PTR = 177774          M.BXNL= 000007        M.1RDT= 000011
FPMODE  000042R     025 F.NIL = 000001        I.STR = 177766          M.BXPP= 000007        M.1REV= 000003
FPP   = 000000          F.NRBD= 000024        KILL  = ****** GX       M.BXPW= 000007        M.1SCN= 000011
FPPLEN= 000004          F.NREC= 000030        LABERR= 104406          M.BXRL= 000007        NAMBUF  003240R     025
FRARGC= ****** GX       F.OPN = 000010        LABLPT  005042R     025 M.BXTS= 000007        NB.DEV= 000200
FREBLK  007400R     025 F.OVBS= 000030        LARGBL  003162R     025 M.BXWA= 000007        NB.DIR= 000100
FRESCT= ****** GX       F.PREV= 000000        LARGCV  003176R     025 M.CEI = 177777        NB.NAM= 000004
FUZZ    000142R     025 F.PRNT= 100000        LARGFL  003160R     025 M.CHR = 000202        NB.SD1= 000400
FXPTR1  003076R     025 F.PROT= 010000        LARGLE  003174R     025 M.CMT = 000206        NB.SD2= 001000
F$EVAL= 000100          F.RACC= 000016        LARGLN  003166R     025 M.DOUT= 000006        NB.SNM= 000040
F$EVI = 000200          F.RATT= 000001        LARGPT  003164R     025 M.DQAD= 000006        NB.STP= 000020
F$EXEC= 000400          F.RCNM= 000034        LARGSF  003172R     025 M.ENCD= 000004        NB.SVR= 000010
F$FUNC= 001000          F.RCTL= 000017        LARGTY  003170R     025 M.EXE = 000000        NB.TYP= 000002
F.ACTL= 000076          F.READ= 004000        LARGVA  003200R     025 M.EXP = 000001        NB.VER= 000001
F.ALOC= 000040          F.REDF= 000040        LBINC   005100R     025 M.FACT= 000001        ND.DEV= 000200
F.BBFS= 000062          F.RES = 000040        LBUFF   003316R     025 M.FIL = 000177        ND.EXT= 000002
F.BDB = 000070          F.RET = 177772        LBUFLN= 000621          M.FLR = 177777        ND.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002        LCOUNT  005106R     025 M.FMT = 000004        ND.PPN= 000100
F.BITS= 177774          F.RTYP= 000000        LENERR= 104412          M.GDUP= 000003        ND.$EX= 000000
F.BKDN= 000026          F.R2  = 000014        LENTAB  000054R     025 M.GDWN= 000003        ND.$ES= 000020
F.BKDS= 000020          F.R3  = 000016        LEXPTR  003100R     025 M.GETI= 000006        ND.$NM= 000000
F.BKEF= 000050          F.R4  = 000020        LFTOP   005066R     025 M.IBM = 000003        ND.$NS= 000040
F.BKP1= 000051          F.SEQN= 000100        LINC    005102R     025 M.IDN = 000000        ND.$PN= 001400
F.BKST= 000024          F.SPDV= 000072        LINE    005016R     025 M.INSP= 000006        NEGSGN  005063R     025
F.BKVB= 000064          F.SPUN= 000074        LININC  005026R     025 M.IOTA= 000000        NINPT1  003302R     025
F.BR  = 000020          F.STBK= 000036        LISIZE= 000020          M.LIT = 000205        NINPT2  003300R     025
F.CHR = 000075          F.STK = 000022        LITOP   005072R     025 M.LN  = 000001        NINSTK  003304R     025
F.CNTG= 000034          F.STOP= 000004        LNLNG = 104541          M.MMD = 000010        NINTMP  003310R     025
F.CODE= 000010          F.STP = 000001        LOCLCT  005036R     025 M.NEG = 000000        NINTYP  003306R     025
F.DFNB= 000046          F.SUSP= 000002        LOWCOR  000002R     025 M.NOT = 000000        NOTAVL= 104472
F.DSPT= 000044          F.TRC = 000002        LSTERP  004766R     025 M.NUB = 000006        NOTOPN= 104473
F.DVNM= 000134          F.UNIT= 000136        LWIDTH  000070R     025 M.NUM = 000204        NPRT1   004762R     025
F.EFBK= 000010          F.URBD= 000020        L.BITS= 000003          M.OUTR= 000012        NULL    007364R     025
F.EFN = 000050          F.VAL = 177776        L.CNT = 000002          M.PITM= 000000        NUMCHN= 000020
F.END = 002000          F.VBN = 000064        L.PRNT= 000001          M.PRDS= 000006        NUMDFL  003230R     025
F.EOBB= 000032          F.VBSZ= 000060        L.PTR = 000000          M.QQAD= 000000        NUMEFL  003233R     025
F.ERR = 000052          GETSM2= ****** GX     L.PTR1= 000004          M.QUAD= 000000        NUMESN  003236R     025
F.FACC= 000043          GNOME   003102R     025 L.SUBS= 000002          M.RAND= 000012        NUMFLG  003230R     025
```

DYDEXP MACRO M1113 08-FEB-80 11:46 PAGE 46-3
SYMBOL TABLE

```
NUMLEN  000062R     025 PRECSN  000052R     025 R.LNUM= 000010          SY.BS  = 000010          T.MON  = 000010
NUMNEG  003232R     025 RANDAC  003064R     025 R.NEXT= 000006          SY.CLR= 000040          T.NIL  = 000004
NUMQFL  003234R     025 RANDV   003062R     025 R.OFF  = 000000          SY.ERP= 000020          T.NUM  = 002000
NUMRIG  003231R     025 RANERR= 104411          R.PREV= 000004          SY.NOU= 000100          T.OPFN= 000001
NUMWID  003226R     025 RARGBL  003204R     025 R.PTR  = 000000          SY.OPN= 000002          T.OTHR= 040000
NUMZFL  003235R     025 RARGCV  003220R     025 R.SEQ  = 000003          SY.PMT= 001000          T.SOP  = 000002
NYIERR= 104432          RARGFL  003202R     025 R.VAR  = 000002          SY.SRT= 000400          T.SPOP= 000040
N.BLSZ= 000036          RARGLE  003216R     025 R6     =%000006          SY.SVC= 000004          T.TMP  = 000200
N.DESC= 000020          RARGLN  003210R     025 R7     =%000007          SY.SVL= 000200          T.VAL  = 100000
N.DEV  = 000032         RARGPT  003206R     025 SI     =%000005          S.AB   = 177776         T.VAR  = 001000
N.DEVN= 000034          RARGSF  003214R     025 SIDAMN= 104433           S.ASB  = 100000         UPPER  = 177400
N.DID  = 000024         RARGTY  003212R     025 SIFRAM  003314R     025 S.BFHD= 000020          UTNAME  000344R     023
N.DVNM= 000032          RARGVA  003222R     025 SISTAK  006350R     025 S.BITS= 000002          VALERR= 104413
N.EXT  = 000014         RBASE   005074R     025 SITOP   006346R     025 S.CHR1= 000003          VARBLE  007400R     025
N.FID  = 000000         RCOUNR  005112R     025 SITP    007144R     025 S.FATT= 000016          VARBLS  000000R     025
N.FNAM= 000006          RCOUNT  005110R     025 SIWARN= 000200           S.FDB  = 000140         VERNUM  000000R     025
N.FTYP= 000014          REDCNT  005070R     025 SIZTAB  000044R     025 S.FLB  = 040000         VRSION= 000002
N.FVER= 000016          REDIGN  005066R     025 SI.BIT= 177774           S.FNAM= 000006          WIDTH   000072R     025
N.NAME= 000006          REDOFF  005072R     025 SI.LEN= 177772           S.FNB  = 000036         WROUND  000020R     025
N.NEXT= 000022          REENTR  003060R     025 SI.RET= 177772           S.FNBW= 000017          WSDATE  000136R     025
N.PPN  = 000024         RESBIT  003142R     025 SI.VAL= 177776           S.FNTY= 000004          WSFULL= 104403
N.STAT= 000020          RESCVT  003152R     025 SKPSIZ  005134R     025 S.FTYP= 000002          WSNAME  000074R     025
N.UNIT= 000034          RESERR= 104440          SPHOLD  000004R     024 S.LEN  = 000002         WSTIME  000132R     025
OBUFLN= 001000          RESLEN  003150R     025 SPINIT  000002R     024 S.LINK= 177774          $FIS   = ****** G
OCHAN   005052R     025 RESPTR  003140R     025 STKINC= 000774           S.NFEN= 000020          $FPP   = ****** G
OMODE   000402R     023 RESSFT  003146R     025 STOP   = 104420         S.TYPE= 140000          $FPPLN= ****** G
OPTMP0  005066R     025 RESTYP  003144R     025 SUBCNT  003104R     025 S.VAL  = 000000         $LDCIF= ****** GX
OPTMP1  005070R     025 RESULT  003136R     025 SUBDCT  003106R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP2  005072R     025 RESVAL  003154R     025 SUBPTR  003110R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP3  005074R     025 RINC    005104R     025 SYMB1   005344R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMF    005340R     025 TMODE   000403R     023 $RT11  = ****** G
OPTMP5  005100R     025 ROTINC  005070R     025 SYMPT1  005346R     025 TMPTAG= 000166          $STF   = ****** GX
OPTMP6  005102R     025 ROUND   000010R     025 SYMTAB  005352R     025 TTYBUF  004140R     025 .BOOLE= 000002
OPTMP7  005104R     025 ROVER   003312R     025 SYMTBF= 104404           TTYLEN= 000622          .CHARA= 000000
OPTMP8  005106R     025 RSTS   = 000000          SYMTOP  005350R     025 TURKEY= 104436          .DEF.  = 000002
OPTMP9  005110R     025 RSX    = 000115          SYMTP   006272R     025 TYPE   = 000166         .FLOAT= 000006
OPTM10  005112R     025 RTOP    005070R     025 SYMWRN= 000054           T.BOL  = 020000         .FSRCB= ****** G
OPTM14  005026R     025 RT11   = 000000          SYNERR= 104407           T.CON  = 004000         .INTEG= 000004
ORIGIN  000030R     025 R.BITS= 000003          SYSERR= 104402           T.DIM  = 000100         .MOLUN  000404R     023
OUTPTR  000066R     025 R.FIX  = 000001          SYSTAT  000000R     024 T.FUN  = 010000         ...GBL= 000000
OVROPR= 000000          R.LCT  = 000002          SY.BAD= 000001           T.LIST= 000400          ...TPC= 000020
PATCH   007344R     025 R.LN   = 000002          SY.BRK= 100000           T.LIT  = 000020

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
```

C 13

```
DYDEXP  MACRO M1113  08-FEB-80 11:46  PAGE 46-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000216      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12958 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:38
DYDEXP,DM1:[305,301]DYDEXP/-SP=APLRX1,APLDEF,DYDEXP,END
```

D 13

```
**FILE**ID**DYDRT
```

```
DDDDDDDD      YY       YY    DDDDDDDD      RRRRRRRR      TTTTTTTTTT
DDDDDDDD      YY       YY    DDDDDDDD      RRRRRRRR      TTTTTTTTTT
DD       DD   YY       YY    DD       DD   RR      RR       TT
DD       DD   YY       YY    DD       DD   RR      RR       TT
DD       DD     YY   YY      DD       DD   RR      RR       TT
DD       DD     YY   YY      DD       DD   RR      RR       TT
DD       DD       YY        DD       DD   RRRRRRRR         TT
DD       DD       YY        DD       DD   RRRRRRRR         TT
DD       DD       YY        DD       DD   RR   RR          TT
DD       DD       YY        DD       DD   RR   RR          TT
DD       DD       YY        DD       DD   RR     RR        TT          ....
DD       DD       YY        DD       DD   RR     RR        TT          ....
DDDDDDDD          YY        DDDDDDDD      RR       RR       TT          ....
DDDDDDDD          YY        DDDDDDDD      RR       RR       TT          ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL            SS              TT
LL              SSSSSS        TT
LL              SSSSSS        TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LL                    SS      TT
LLLLLLLLLL    SSSSSSSS        TT
LLLLLLLLLL    SSSSSSSS        TT
```



```
MAIN - MACRO M1113  08-FEB-80 11:48
```

.MAIN. MACRO M1113 08-FEB-80 11:48
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    CONVERSION       DYADIC LEFT, TO FLOAT
```

DYDRT MACRO M1113 08-FEB-80 11:48
TABLE OF CONTENTS

45-	62	CONVERSION	DYADIC LEFT, TO BOOLEAN
45-	106	CONVERSION	DYADIC LEFT, TO INTEGER
45-	142	CONVERSION	DYADIC LEFT, FROM CHARACTER
46-	159	CONVERSION	DYADIC RIGHT, TO FLOAT
47-	199	CONVERSION	DYADIC RIGHT, TO INTEGER
47-	233	CONVERSION	DYADIC RIGHT, TO BOOLEAN
47-	270	CONVERSION	DYADIC RIGHT, FROM CHARACTER

DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 1

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 1


                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M

                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP

                000004          FPPLEN  =  4.           ;SINGLE PRECISION

                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 2
APLDEF

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```


APL MACRO M1113 08-FEB-80 11:48 PAGE 42

```
APL     MACRO M1113  08-FEB-80 11:48  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  DYDRT
                                .IDENT  \V02.03\

                                .SBTTL  CONVERSION      DYADIC LEFT, TO FLOAT

                                ;
                                ; <BOOLEAN>   => AC1            (FLOATING POINT)
                                ;
 000000                                 .PSECT  CODE

 000000                         FDL.BF::
 000000                                 CLRF    AC1             ;CLEAR AC1
 000006  105712                         TSTB    (R2)            ;CHECK SOURCE
 000010  001404                         BEQ     1$              ;IT IS ZERO
 000012                                 LDF     #^F1.0,AC1      ;LOAD A 1.
 000022  063702  003160'        1$:     ADD     LARGFL,R2       ;INCREMENT POINTER
 000026                                 RETURN                  ;FINISHED

                                ;
                                ; <INTEGER>   => AC1            (FLOATING POINT)
                                ;
 000030                         FDL.IF::
 000030                                 LDCIF   (R2),AC1        ;GET SOURCE
 000036  063702  003160'                ADD     LARGFL,R2       ;INCREMENT
 000042                                 RETURN                  ;FINISHED

                                ;
                                ; <FLOAT>     => AC1            (FLOATING POINT)
                                ;
 000044                         FDL.FF::
 000044                                 LDF     (R2),AC1        ;GET SOURCE
 000052  063702  003160'                ADD     LARGFL,R2       ;ADD TO POINTER
 000056                                 RETURN                  ;FINISHED
```

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 45
CONVERSION      DYADIC LEFT, TO BOOLEAN

                                        .SBTTL  CONVERSION      DYADIC LEFT, TO BOOLEAN
                                        ;
                                        ; <BOOLEAN>  => LARGVA          (BOOLEAN)
                                        ;       (This also works for Character)
                                        ;
000060                                  .PSECT  CODE

000060                                  FDL.CC::
000060                                  FDL.BB::
000060  111237  003200'                         MOVB    (R2),LARGVA     ;TRANSFER
000064  105037  003201'                         CLRB    LARGVA+1        ;Clear upper byte so compares are unsigned
000070  063702  003160'                         ADD     LARGFL,R2       ;INCREMENT POINTER
000074                                          RETURN

                                        ;
                                        ; <INTEGER>  => LARGVA          (BOOLEAN)
                                        ;
000076                                  FDL.IB::
000076  011237  003200'                         MOV     (R2),LARGVA     ;TRANSFER
000102  032737  177776  003200'                 BIT     #^C1,LARGVA     ;HOW ARE THE BITS?
000110  001003                                  BNE     1$              ;WON'T WORK
000112  063702  003160'                         ADD     LARGFL,R2       ;INCREMENT POINTER
000116                                          RETURN
000120  104417                          1$:     DOMERR                  ;DOMAIN ERROR

                                        ;
                                        ; <FLOAT>    => LARGVA          (BOOLEAN)
                                        ;
000122                                  FDL.FB::
000122                                          LDF     (R2),AC0        ;GET DATA
000130                                          CALL    F.TO.I          ;CONVERT
000134  104417                          1$:     DOMERR                  ;ERROR
000136  032737  177776  003154'                 BIT     #^C1,RESVAL     ;ANY HIGH BITS SET?
000144  001373                                  BNE     1$              ;YES, ERROR
000146  013737  003154' 003200'                 MOV     RESVAL,LARGVA   ;TRANFER IT
000154  063702  003160'                         ADD     LARGFL,R2       ;INCREMENT POINTER
000160                                          RETURN

                                        .SBTTL  CONVERSION      DYADIC LEFT, TO INTEGER
                                        ;
                                        ; <BOOLEAN>  => LARGVA          (INTEGER)
                                        ;
000162                                  FDL.BI::
000162  005037  003200'                         CLR     LARGVA          ;ASSUME 0
000166  105712                                  TSTB    (R2)            ;IS IT ZERO?
000170  001402                                  BEQ     1$              ;YES
000172  005237  003200'                         INC     LARGVA          ;NO, IT'S A 1
000176  063702  003160'                 1$:     ADD     LARGFL,R2       ;INCREMENT POINTER
000202                                          RETURN

                                        ;
                                        ; <INTEGER>  => LARGVA          (INTEGER)
                                        ;
000204                                  FDL.II::
000204  011237  003200'                         MOV     (R2),LARGVA     ;TRANSFER
000210  063702  003160'                         ADD     LARGFL,R2       ;INCREMENT POINTER
```

L 13

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 45-1
CONVERSION      DYADIC LEFT, TO INTEGER

    124 000214                                          RETURN
    125
    126                                         ;
    127                                         ; <FLOAT>   => LARGVA           (INTEGER)
    128                                         ;
    129 000216                                  FDL.FI::
    131 000216                                          LDF     (R2),AC0        ;GET FP NUMBER
    136 000224                                          CALL    F.TO.I          ;CONVERT
    137 000230  104417                                  DOMERR                  ;DOMAIN ERROR
    138 000232  013737  003154' 003200'                 MOV     RESVAL,LARGVA   ;TRANSFER IT
    139 000240  063702  003160'                         ADD     LARGFL,R2       ;INCREMENT POINTER
    140 000244                                          RETURN                  ;   AND RETURN
    141
    142                                         .SBTTL  CONVERSION      DYADIC LEFT, FROM CHARACTER
    143                                         ;
    144                                         ; <CHARACTER>   => LARGVA       (BOOLEAN, INTEGER, FLOATING)
    145                                         ;
    146 000246                                  FDL.CB::
    147 000246                                  FDL.CI::
    148 000246                                  FDL.CF::
    149 000246  020427  000000G                         CMP     R4,#OP.EQU      ;Is the function 'Equal''
    150 000252  001404                                  BEQ     1$              ;Yes
    151 000254  020427  000000G                         CMP     R4,#OP.NEQ      ;Is the function 'Not Equal''
    152 000260  001401                                  BEQ     1$              ;Yes
    153 000262  104417                                  DOMERR                  ;No, Then Domain error
    154 000264  005037  003200'         1$:             CLR     LARGVA          ;Make it zero just in case
    155 000270                                          CLRF    AC1             ;Floating one too
    156 000276  063702  003160'                         ADD     LARGFL,R2       ;Increment the argument pointer
    157 000302                                          RETURN
```

M 13

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 46
```

```
                                .SBTTL  CONVERSION      DYADIC RIGHT, TO FLOAT

                                ;
                                ; <BOOLEAN>   => AC2              (FLOATING POINT)
                                ;
000304                                  .PSECT  CODE

000304                          FDR.BF::
000304                                  CLRF    AC2             ;CLEAR AC2
000312  105713                         TSTB    (R3)            ;CHECK SOURCE
000314  001404                         BEQ     1$              ;IT IS ZERO
000316                                  LDF     #^F1.0,AC2      ;LOAD A 1.
000326  063703  003202'         1$:    ADD     RARGFL,R3       ;INCREMENT POINTER
000332                                  RETURN                  ;FINISHED

                                ;
                                ; <INTEGER>   => AC2              (FLOATING POINT)
                                ;
000334                          FDR.IF::
000334                                  LDCIF   (R3),AC2        ;GET SOURCE
000342  063703  003202'                 ADD     RARGFL,R3       ;INCREMENT POINTER
000346                                  RETURN                  ;FINISHED

                                ;
                                ; <FLOAT>     => AC2              (FLOATING POINT)
                                ;
000350                          FDR.FF::
000350                                  LDF     (R3),AC2        ;GET SOURCE
000356  063703  003202'                 ADD     RARGFL,R3       ;INCREMENT POINTER
000362                                  RETURN                  ;FINISHED
```

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 47
CONVERSION      DYADIC RIGHT, TO INTEGER

                                        .SBTTL  CONVERSION      DYADIC RIGHT, TO INTEGER

                                        ;
                                        ; <BOOLEAN>  => RARGVA          (INTEGER)
                                        ;
000364                                          .PSECT  CODE

000364                                  FDR.BI::
000364  005037  003222'                         CLR     RARGVA          ;ASSUME 0
000370  105713                                 TSTB    (R3)            ;IS IT ZERO?
000372  001402                                 BEQ     1$              ;YES
000374  005237  003222'                         INC     RARGVA          ;NO, SET IT TO 1
000400  063703  003202'                 1$:     ADD     RARGFL,R3       ;INCREMENT POINTER
000404                                          RETURN

                                        ;
                                        ; <INTEGER>  => RARGVA          (INTEGER)
                                        ;
000406                                  FDR.II::
000406  011337  003222'                         MOV     (R3),RARGVA     ;COPY DATA
000412  063703  003202'                         ADD     RARGFL,R3       ;INCREMENT POINTER
000416                                          RETURN

                                        ;
                                        ; <FLOAT>    => RARGVA          (INTEGER)
                                        ;
000420                                  FDR.FI::
000420                                          LDF     (R3),AC0        ;FETCH ARG
000426                                          CALL    F.TO.I          ;CONVERT TO INTEGER
000432  104417                                  DOMERR                  ;DIDN'T GO
000434  013737  003154' 003222'                 MOV     RESVAL,RARGVA   ;COPY IT
000442  063703  003202'                         ADD     RARGFL,R3       ;INCREMENT POINTER
000446                                          RETURN

                                        .SBTTL  CONVERSION      DYADIC RIGHT, TO BOOLEAN
                                        ;
                                        ; <BOOLEAN>  => RARGVA          (BOOLEAN)
                                        ;       (This also works for Character)
                                        ;
000450                                  FDR.BC::                ;Special fudge for ''='' reduction
000450                                  FDR.CC::
000450                                  FDR.BB::
000450  111337  003222'                         MOVB    (R3),RARGVA     ;TRANSFER
000454  105037  003223'                         CLRB    RARGVA+1        ;Clear upper byte so compares are unsigned
000460  063703  003202'                         ADD     RARGFL,R3       ;INCREMENT POINTER
000464                                          RETURN

                                        ;
                                        ; <INTEGER>  => RARGVA          (BOOLEAN)
                                        ;
000466                                  FDR.IB::
000466  011337  0032c2'                         MOV     (R3),RARGVA     ;TRANSFER
000472  032737  177776  003222'                 BIT     #^C1,RARGVA     ;CHECK HIGH BITS
000500  001003                                 BNE     1$              ;NOPE, DIDN'T WORK
000502  063703  003202'                         ADD     RARGFL,R3       ;INCREMENT POINTER
000506                                          RETURN
000510  104417                          1$:     DOMERR                  ;WON'T CONVERT
```

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 47-1
CONVERSION      DYADIC RIGHT, TO BOOLEAN

                                        ;
                                        ; <FLOAT>     => RARGVA           (BOOLEAN)
                                        ;
                                        FDR.FB::
000512                                          LDF     (R3),AC0            ;GET THE VALUE
000512                                          CALL    F.TO.I              ;CONVERT IT
000520  104417                          1$:     DOMERR                      ;ROUND PEG IN SQUARE HOLE
000524  032737  177776  003154'                BIT     #^C1,RESVAL         ;CHECK THOSE BITS
000526  001373                                  BNE     1$                  ;CANNOT FORCE IT
000534  013737  003154' 003222'                MOV     RESVAL,RARGVA       ;COPY TO CORRECT PLACE
000536  063703  003202'                        ADD     RARGFL,R3           ;INCREMENT POINTER
000544                                          RETURN

                                        .SBTTL  CONVERSION      DYADIC RIGHT, FROM CHARACTER
                                        ;
                                        ; <CHARACTER>   => RARGVA           (BOOLEAN, INTEGER, FLOATING)
                                        ;
                                        FDR.CB::
                                        FDR.CI::
                                        FDR.CF::
000552  020427  000000G                          CMP     R4,#OP.EQU          ;Is the function 'Equal''
000556  001404                                  BEQ     1$                  ;Yes
000560  020427  000000G                          CMP     R4,#OP.NEQ          ;Is the function 'Not Equal''
000564  001401                                  BEQ     1$                  ;Yes
000566  104417                                  DOMERR                      ;No, Then Domain Error
000570  005037  003222'                1$:     CLR     RARGVA              ;Make it zero just in case
000574                                          CLRF    AC2                 ;Floating one too
000602  063703  003202'                        ADD     RARGFL,R3           ;Increment argument pointer
000606                                          RETURN
```



```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 48
```

```
    287 000000                                      .PSECT  TABLES
    288                                             .EVEN
    289
    290 000000                              LARGCT::
    291 000000                              CLARGC::
    292 000000  000060' 000000G 000000G             .WORD   FDL.CC, DOMER,  DOMER,  DOMER
        000006  000000G
    293 000010                              BLARGC::
    294 000010  000246' 000060' 000076'             .WORD   FDL.CB, FDL.BB, FDL.IB, FDL.FB
        000016  000122'
    295 000C20                              ILARGC::
    296 000020  000246' 000162' 000204'             .WORD   FDL.CI, FDL.BI, FDL.II, FDL.FI
        000026  000216'
    297 000030                              FLARGC::
    298 000030  000246' 000000' 000030'             .WORD   FDL.CF, FDL.BF, FDL.IF, FDL.FF
        000036  000044'
    299
    300 000040                              RARGCT::
    301 000040                              CRARGC::
    302 000040  000450' 000450' 000000G             .WORD   FDR.CC, FDR.BC, DOMER,  DOMER
        000046  000000G
    303 000050                              BRARGC::
    304 000050  000552' 000450' 000466'             .WORD   FDR.CB, FDR.BB, FDR.IB, FDR.FB
        000056  000512'
    305 000060                              IRARGC::
    306 000060  000552' 000364' 000406'             .WORD   FDR.CI, FDR.BI, FDR.II, FDR.FI
        000066  000420'
    307 000070                              FRARGC::
    308 000070  000552' 000304' 000334'             .WORD   FDR.CF, FDR.BF, FDR.IF, FDR.FF
        000076  000350'
```

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 50
CONVERSION      DYADIC RIGHT, FROM CHARACTER

      1         000001                              .END
```



```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 50-1
```

DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 50-1
SYMBOL TABLE

```
AC0    =%000000          CHRFIL= 000177          CTMAX   000212R  025 D.COMP= 000001          EDICHR  005063R  025
AC1    =%000001          CHRFRO= 000135          CTRMAX  000172R  025 D.CONT= 000006          EDICNT  005064R  025
AC2    =%000002          CHRLBR= 000073          C.ALOC= 000162      D.CTLM= 000000          EDIFLG  005062R  025
AC3    =%000003          CHRLPR= 000072          C.BLOK= 000064      D.DEAL= 000003          EDIT   = 000003
AC4    =%000004          CHRMNS= 000137          C.BLSZ= 000140      D.DECD= 000002          EIS    = 000000
AC5    =%000005          CHRMUL= 000075          C.BUFF= 177760      D.DIV = 000000          ELESIZ  005132R  025
ALLINT  003274R  025 CHRNEG= 000100          C.BUFL= 177762      D.DMD = 000010          ENDCOR  000004R  025
ARYRET  003276R  025 CHRPDL= 000271          C.CCNT= 177766      D.DQAD= 177777          EOF    = 104475
ASCQUO= 000042          CHRPLU= 000055          C.CPTR= 177764      D.DROP= 000002          ERRMAX= 000166
A.ALEN= 000004          CHRPWR= 000120          C.CZ   = 001000     D.ENCD= 000002          ERRNUM  004772R  025
A.ARG  = 000040         CHRQST= 000121          C.DIG  = 000040     D.EQ   = 000000         ERRPTR  004770R  025
A.ASB  = 000200         CHRQUO= 000113          C.FASW= 020000      D.EXPD= 000002          ERR0    004774R  025
A.BITS= 000003          CHRRBR= 000047          C.FLEN= 000010      D.FGET= 000005          EXEBUF  005054R  025
A.CON  = 000020         CHRRPR= 000042          C.FLSH= 004000      D.FMT  = 000004         E$CT    005056R  025
A.DIMS= 000002          CHRSEM= 000074          C.FRC  = 000001     D.FPNT= 000005          E$PT    005060R  025
A.DIM1= 000006          CHRSMO= 000112          C.IDEN= 000010      D.FPUT= 000005          FA.APD= 000100
A.NUM  = 000006         CHRSPA= 000040          C.NAME= 000102      D.GE   = 000000         FA.CRE= 000010
A.PTR  = 000000         CHRUPA= 000137          C.NOBF= 000400      D.GT   = 000000         FA.DLK= 001000
A.TMP  = 000010         CHR.BS= 000010          C.NUM  = 000020     D.IBM  = 000002         FA.ENB= 100000
BADDIR= 104463          CHR.CC= 000003          C.OP   = 000004     D.IND  = 000002         FA.EXC= 002000
BADERR= 104437          CHR.CR= 000015          C.OPN  = 010000     D.INNR= 000013          FA.EXT= 000004
BLARGC  000010RG  002 CHR.CZ= 000032          C.RDLN= 177776      D.INTR= 000006          FA.NSP= 000100
BLZBUB  005116R  025 CHR.GT= 000046          C.REWT= 002000      D.LE   = 000000         FA.NSR= 000200
BRARGC  000050RG  002 CHR.LF= 000012          C.RND  = 000010     D.LOG  = 000001         FA.POS= 010000
BRSELF  005046R  025 CHR.LT= 000043          C.SBIT= 177756      D.LT   = 000000         FA.RD  = 000001
CCNT    005072R  025 CHR.SI= 000017          C.SPFX= 000100      D.MAX  = 000000         FA.RWD= 004000
CHANEX  003060R  025 CHR.SL= 000057          C.SPGL= 000200      D.MEM  = 000002         FA.SEQ= 040000
CHAN00  000256R  025 CHR.SO= 000016          C.SPLX= 000001      D.MIN  = 000000         FA.SHR= 000040
CHAN01  000330R  025 CHR.TB= 000011          C.SPRS= 000002      D.MUL  = 000000         FA.TMP= 000020
CHAN02  000512R  025 CHR.UL= 000106          C.SRC  = 177774     D.NE   = 000000         FA.WCK= 020000
CHAN03  000674R  025 CHR..A= 000141          C.SRCP= 000024      D.NN   = 000000         FA.WRT= 000002
CHAN04  001056R  025 CHR..E= 000145          C.TT   = 000004     D.NR   = 000000         FDL.BB  000060RG  020
CHAN05  001240R  025 CHR..S= 000163          C.WIDC= 177770      D.OR   = 000000         FDL.BF  000000RG  020
CHAN06  001422R  025 CHR..T= 000164          C.WIDR= 177772      D.PLUS= 000000          FDL.BI  000162RG  020
CHAN07  001604R  025 CHR..U= 000165          DEBUG  = 000000     D.PSUB= 000006          FDL.CB  000246RG  020
CHAN08  001766R  025 CHR..Z= 000172          DEFERR= 104405      D.PWR  = 000001         FDL.CC  000060RG  020
CHAN09  002150R  025 CHR..0= 000060          DEVIL   005114R  025 D.QQAD= 000000          FDL.CF  000246RG  020
CHAN10  002332R  025 CHR..1= 000061          DEVNTF= 104470      D.QUAD= 000000          FDL.CI  000246RG  020
CHAN11  002514R  025 CH.AND= 000001          DIGITS  000006R  025 D.RES  = 000000          FDL.FB  000122RG  020
CHAN12  002676R  025 CLARGC  000000RG  002 DIMAX  = 000020      D.RHO  = 000000         FDL.FF  000044RG  020
CHAN13  000022R  023 CMDBUF  003250R  025 DIMBLK  003112R  025 D.ROT  = 000003         FDL.FI  000216RG  020
CHAN14  000204R  023 CMDMAX= 000010          DIMPTR  003114R  025 D.SCON= 000006          FDL.IB  000076RG  020
CHNOPN= 104471          CNTSIZ  005130R  025 DIMTYP  003116R  025 D.SDIF= 000006          FDL.IF  000030RG  020
CHRAND= 000051          COMBIN  005076R  025 DIRFUL= 104466      D.SUB  = 000000         FDL.II  000204RG  020
CHRBEL= 000007          COMCNT  005126R  025 DOMER  = ****** GX  D.SUBS= 000006          FDR.BB  000450RG  020
CHRBKR= 000133          COMERR= 104425          DOMERR= 104417      D.TAKE= 000002          FDR.BC  000450RG  020
CHRBOL= 000200          COMPTR  005122R  025 DPTERR= 104416      D.TRAN= 000004          FDR.BF  000304RG  020
CHRBOX= 000114          COMRTN  003156R  025 DYDCNT  003224R  025 D.UNON= 000006          FDR.BI  000364RG  020
CHRCMA= 000054          COMSKP  005124R  025 D.AND  = 000000      D.$MAX= 000000          FDR.CB  000552RG  020
CHRCOL= 000076          CORPTR  005120R  025 D.BXCT= 000007      D.$MIN= 000000          FDR.CC  000450RG  020
CHRDEL= 000107          CRARGC  000040RG  002 D.BXIO= 000007      D.$MUL= 000000          FDR.CF  000552RG  020
CHRDIV= 000053          CS.ASS= 000200          D.BXNL= 000007      D.$PLU= 000000          FDR.CI  000552RG  020
CHRDLR= 000176          CS.BOO= 000010          D.BXPP= 000007      D.$RES= 000000          FDR.FB  000512RG  020
CHRDLT= 000110          CS.FLO= 000020          D.BXPW= 000007      D.$SUB= 000000          FDR.FF  000350RG  020
CHRDOT= 000056          CS.IDN= 000040          D.BXRL= 000007      D.1COM= 000001          FDR.FI  000420RG  020
CHREP  = 000105         CS.ONE= 000100          D.CIRC= 000001      D.1EXP= 000002          FDR.IB  000466RG  020
CHRERR= 104431          CTEPS   000152R  025 D.COMB= 000001      D.1ROT= 000003          FDR.IF  000334RG  020
```


DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 50-2

DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 50-2
SYMBOL TABLE

```
FDR.II  000406RG    020 F$EVAL= 000100          F.RACC= 000016          LARGFL  003160R     025 M.CHR = 000202
FD.BLK= 000010          F$EVI = 000200          F.RATT= 000001          LARGLE  003174R     025 M.CMT = 000206
FD.CCL= 000002          F$EXEC= 000400          F.RCNM= 000034          LARGLN  003166R     025 M.DOUT= 000006
FD.COM= 020000          F$FUNC= 001000          F.RCTL= 000017          LARGPT  003164R     025 M.DQAD= 000006
FD.CR = 000002          F.ACTL= 000076          F.READ= 004000          LARGSF  003172R     025 M.ENCD= 000004
FD.DIR= 000010          F.ALOC= 000040          F.REDF= 000040          LARGTY  003170R     025 M.EXE = 000000
FD.FTN= 000001          F.BBFS= 000062          F.RES = 000040          LARGVA  003200R     025 M.EXP = 000001
FD.F11= 040000          F.BDB = 000070          F.RET = 177772          LBINC   005100R     025 M.FACT= 000001
FD.INS= 000010          F.BGBC= 000057          F.RSIZ= 000002          LBUFF   003316R     025 M.FIL = 000177
FD.ISP= 002000          F.BITS= 177774          F.RTYP= 000000          LBUFLN= 000621          M.FLR = 177777
FD.MNT= 100000          F.BKDN= 000026          F.R2  = 000014          LCOUNT  005106R     025 M.FMT = 000004
FD.OSP= 004000          F.BKDS= 000020          F.R3  = 000016          LENERR= 104412          M.GDUP= 000003
FD.PLC= 000004          F.BKEF= 000050          F.R4  = 000020          LENTAB  000054R     025 M.GDWN= 000003
FD.PRN= 000004          F.BKP1= 000051          F.SEQN= 000100          LEXPTR  003100R     025 M.GETI= 000006
FD.PSE= 010000          F.BKST= 000024          F.SPDV= 000072          LFTOP   005066R     025 M.IBM = 000003
FD.RAH= 000001          F.BKVB= 000064          F.SPUN= 000074          LINC    005102R     025 M.IDN = 000000
FD.RAN= 000002          F.BR  = 000020          F.STBK= 000036          LINE    005016R     025 M.INSP= 000006
FD.REC= 000001          F.CHR = 000075          F.STK = 000022          LININC  005026R     025 M.IOTA= 000000
FD.RWM= 000001          F.CNTG= 000034          F.STOP= 000004          LISIZE= 000020          M.LIT = 000205
FD.SDI= 000020          F.CODE= 000010          F.STP = 000001          LITOP   005072R     025 M.LN  = 000001
FD.SQD= 000040          F.DFNB= 000046          F.SUSP= 000002          LNLNG = 104541          M.MMD = 000010
FD.TTY= 000004          F.DSPT= 000044          F.TO.I= ****** GX       LOCLCT  005036R     025 M.NEG = 000000
FD.WBH= 000002          F.DVNM= 000134          F.TRC = 000002          LOWCOR  000002R     025 M.NOT = 000000
FF.CHR= 000005          F.EFBK= 000010          F.UNIT= 000136          LSTERP  004766R     025 M.NUB = 000006
FF.NV = 000003          F.EFN = 000050          F.URBD= 000020          LWIDTH  000070R     025 M.NUM = 000204
FF.POE= 000002          F.END = 002000          F.VAL = 177776          L.BITS= 000003          M.OUTR= 000012
FF.RWD= 000001          F.EOBB= 000032          F.VBN = 000064          L.CNT = 000002          M.PITM= 000000
FF.RWF= 000006          F.ERR = 000052          F.VBSZ= 000060          L.PRNT= 000001          M.PRDS= 000006
FF.SPC= 000004          F.FACC= 000043          GNOME   003102R     025 L.PTR = 000000          M.QQAD= 000000
FHEADR  005012R     025 F.FFBY= 000014          HASHC   005136R     025 L.PTR1= 000004          M.QUAD= 000000
FHRET   005040R     025 F.FLAG= 000002          HASHTB  005140R     025 L.SUBS= 000002          M.RAND= 000012
FILNTF= 104467          F.FLB = 000100          HSHMSK= 177601          MACHIN= 000106          M.RAVL= 000000
FIS   = 000000          F.FNAM= 000110          HSHSIZ= 000200          MAXCHN= 000014          M.RCIP= 000000
FISCON= 046000          F.FNB = 000102          ICHAN   005050R     025 MONBLK  003120R     025 M.RDCT= 000011
FIXPTR  003074R     025 F.FNDF= 000010          ILARGC  000020RG    002 MONCVT  003132R     025 M.REVS= 000003
FLAGS   004764R     025 F.FTYP= 000116          ILFNAM= 104464          MONLEN  003130R     025 M.ROLL= 000000
FLARGC  000030RG    002 F.FUN = 000010          ILLCHN= 104540          MONPTR  003122R     025 M.SATA= 000001
FLCBUF  003250R     025 F.FVER= 000120          ILLSW = 104565          MONSFT  003126R     025 M.SCAN= 000011
FLINE   005014R     025 F.HEAD= 000006          INDERR= 104410          MONTYP  003124R     025 M.SHAP= 000000
FLOEXP  003270R     025 F.HIBK= 000004          INNTMP  003250R     025 MONVAL  003134R     025 M.SIGN= 000000
FLONEG  003272R     025 F.LBL = 000020          INNTM2  005016R     025 M.ABS = 000000          M.SUBR= 000001
FLSWTB  000222R     025 F.LINE= 000012          INPTR   000064R     025 M.BOL = 000200          M.SYM = 000203
FLTLEN= 000010          F.LOC = 000004          INSNAM= 104462          M.BXAV= 000007          M.SYN = 000201
FNINSI  005044R     025 F.LSTC= 000004          IOERR = 104477          M.BXCR= 000007          M.TRAN= 000012
FORIGN  000032R     025 F.LUN = 000042          IRARGC  000060RG    002 M.BXCT= 000007          M.$ABS= 000000
FO.APD= 000106          F.MBCT= 000054          I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FO.MFY= 000002          F.MBC1= 000055          I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FO.RD = 000001          F.MBFG= 000056          I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.UPD= 000006          F.MON = 000002          I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.WRT= 000016          F.NIL = 000001          I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FPMODE  000042R     025 F.NRBD= 000024          I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FPP   = 000000          F.NREC= 000030          I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FPPLEN= 000004          F.OPN = 000010          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FRARGC  000070RG    002 F.OVBS= 000030          LABLPT  005042R     025 M.BXRL= 000007          NAMBUF  003240R     025
FREBLK  007400R     025 F.PREV= 000000          LARGBL  003162R     025 M.BXTS= 000007          NB.DEV= 000200
FUZZ    000142R     025 F.PRNT= 100000          LARGCT  000000RG    002 M.BXWA= 000007          NB.DIR= 000100
FXPTR1  003076R     025 F.PROT= 010000          LARGCV  003176R     025 M.CEI = 177777          NB.NAM= 000004
```


DYDRT MACRO M1113 08-FEB-80 11:48 PAGE 50-3

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 50-3
SYMBOL TABLE

NB.SD1= 000400          N.NAME= 000006          RESERR= 104440          SYMB1   005344R 025 T.BOL = 020000
NB.SD2= 001000          N.NEXT= 000022          RESLEN  003150R 025 SYMF  005340R 025 T.CON = 004000
NB.SNM= 000040          N.PPN = 000024          RESPTR  003140R 025 SYMPT1 005346R 025 T.DIM = 000100
NB.STP= 000020          N.STAT= 000020          RESSFT  003146R 025 SYMTAB 005352R 025 T.FUN = 010000
NB.SVR= 000010          N.UNIT= 000034          RESTYP  003144R 025 SYMTBF= 104404       T.LIST= 000400
NB.TYP= 000002          OBUFLN= 001000          RESULT  003136R 025 SYMTOP 005350R 025 T.LIT = 000020
NB.VER= 000001          OCHAN   005052R 025 RESVAL  003154R 025 SYMTP  006272R 025 T.MON = 000010
ND.DEV= 000200          OMODE   000402R 023 RINC    005104R 025 SYMWRN= 000054       T.NIL = 000004
ND.EXT= 000002          OPTMP0  005066R 025 RLTBAS  005066R 025 SYNERR= 104407       T.NUM = 002000
ND.NAM= 000004          OPTMP1  005070R 025 ROTINC  005070R 025 SYSERR= 104402       T.OPFN= 000001
ND.PPN= 000100          OPTMP2  005072R 025 ROUND   000010R 025 SYSTAT  000000R 024 T.OTHR= 040000
ND.SEX= 000000          OPTMP3  005074R 025 ROVER   003312R 025 SY.BAD= 000001       T.SOP = 000002
ND.SES= 000020          OPTMP4  005076R 025 RSTS  = 000000      SY.BRK= 100000       T.SPOP= 000040
ND.SNM= 000000          OPTMP5  005100R 025 RSX   = 000115      SY.BS = 000010       T.TMP = 000200
ND.SNS= 000040          OPTMP6  005102R 025 RTOP    005070R 025 SY.CLR= 000040       T.VAL = 100000
ND.SPN= 001400          OPTMP7  005104R 025 RT11  = 000000      SY.ERP= 000020       T.VAR = 001000
NEGSGN  005063R 025 OPTMP8  005106R 025 R.BITS= 000003      SY.NOU= 000100       UPPER = 177400
NINPT1  003302R 025 OPTMP9  005110R 025 R.FIX = 000001      SY.OPN= 000002       UTNAME  000344R 023
NINPT2  003300R 025 OPTM10  005112R 025 R.LCT = 000002      SY.PMT= 001000       VALERR= 104413
NINSTK  003304R 025 OPTM14  005026R 025 R.LN  = 000002      SY.SRT= 000400       VARBLE  007400R 025
NINTMP  003310R 025 OP.EQU= ****** GX       R.LNUM= 000010      SY.SVC= 000004       VARBLS  000000R 025
NINTYP  003306R 025 OP.NEQ= ****** GX       R.NEXT= 000006      SY.SVL= 000200       VERNUM  000000R 025
NOTAVL= 104472          ORIGIN  000030R 025 R.OFF = 000000      S.AB  = 177776       VRSION= 000002
NOTOPN= 104473          OUTPTR  000066R 025 R.PREV= 000004      S.ASB = 100000       WIDTH   000072R 025
NPRT1   004762R 025 OVROPR= 000000      R.PTR = 000000      S.BFHD= 000020       WROUND  000020R 025
NULL    007364R 025 PATCH   007344R 025 R.SEQ = 000003      S.BITS= 000002       WSDATE  000136R 025
NUMCHN= 000020          PRECSN  000052R 025 R.VAR = 000002      S.CHR1= 000003       WSFULL= 104403
NUMDFL  003230R 025 RANDAC  003064R 025 R6    =%000006      S.FATT= 000016       WSNAME  000074R 025
NUMEFL  003233R 025 RANDY   003062R 025 R7    =%000007      S.FDB = 000140       WSTIME  000132R 025
NUMESN  003236R 025 RANERR= 104411      SI    =%000005      S.FLB = 040000       $CLRF = ****** GX
NUMFLG  003230R 025 RARGBL  003204R 025 SIDAMN= 104433      S.FNAM= 000006       $FIS  = ****** G
NUMLEN  000062R 025 RARGCT  000040RG 002 SIFRAM  003314R 025 S.FNB = 000036      $FPP  = ****** G
NUMNEG  003232R 025 RARGCV  003220R 025 SISTAK  006350R 025 S.FNBW= 000017      $FPPLN= ****** G
NUMQFL  003234R 025 RARGFL  003202R 025 SITOP   006346R 025 S.FNTY= 000004      $LDCIF= ****** GX
NUMRIG  003231R 025 RARGLE  003216R 025 SITP    007144R 025 S.FTYP= 000002      $LDF  = ****** GX
NUMWID  003226R 025 RARGLN  003210R 025 SIWARN= 000200      S.LEN = 000002       $MACHN= ****** G
NUMZFL  003235R 025 RARGPT  003206R 025 SIZTAB  000044R 025 S.LINK= 177774      $RSTS = ****** G
NYIERR= 104432          RARGSF  003214R 025 SI.BIT= 177774      S.NFEN= 000020       $RSX  = ****** G
N.BLSZ= 000036          RARGTY  003212R 025 SI.LEN= 177772      S.TYPE= 140000       $RT11 = ****** G
N.DESC= 000020          RARGVA  003222R 025 SI.RET= 177772      S.VAL = 000000       .BOOLE= 000002
N.DEV = 000032          RBASE   005074R 025 SI.VAL= 177776      TKDIMS  005072R 025 .CHARA= 000000
N.DEVN= 000034          RCOUNR  005112R 025 SKPSIZ  005134R 025 TKFRAM  005066R 025 .DEF. = 000002
N.DID = 000024          RCOUNT  005110R 025 SPHOLD  000004R 024 TKLEFT  005070R 025 .FLOAT= 000006
N.DVNM= 000032          REDCNT  005070R 025 SPINIT  000002R 024 TMODE   000403R 023 .FSRCB= ****** G
N.EXT = 000014          REDIGN  005066R 025 STKINC= 000774      TMPTAG= 000166       .INTEG= 000004
N.FID = 000000          REDOFF  005072R 025 STOP  = 104420      TTYBUF  004140R 025 .MOLUN  000404R 023
N.FNAM= 000006          REENTR  003060R 025 SUBCNT  003104R 025 TTYLEN= 000622       ...GBL= 000000
N.FTYP= 000014          RESBIT  003142R 025 SUBDCT  003106R 025 TURKEY= 104436       ...TPC= 000020
N.FVER= 000016          RESCVT  003152R 025 SUBPTR  003110R 025 TYPE  = 000166

. ABS.  000000  000
        000000  001
TABLES  000100  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
```

H 14

```
DYDRT   MACRO M1113  08-FEB-80 11:48  PAGE 50-4
SYMBOL TABLE

DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000610      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13160 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:00
DYDRT,DM1:[305,301]DYDRT/-SP=APLRX1,APLDEF,DYDRT,END
```

I 14

```
**FILE**ID**ECHR
```

```
**FILE**ID**ECHR
```

```
EEEEEEEEEE      CCCCCCCC    HH      HH    RRRRRRRR
EEEEEEEEEE      CCCCCCCC    HH      HH    RRRRRRRR
EE            CC            HH      HH    RR      RR
EE            CC            HH      HH    RR      RR
EE            CC            HH      HH    RR      RR
EE            CC            HH      HH    RR      RR
EEEEEEEE      CC            HHHHHHHHHH    RRRRRRRR
EEEEEEEE      CC            HHHHHHHHHH    RRRRRRRR
EE            CC            HH      HH    RR  RR
EE            CC            HH      HH    RR  RR
EE            CC            HH      HH    RR    RR          ....
EE            CC            HH      HH    RR    RR          ....
EEEEEEEEEE      CCCCCCCC    HH      HH    RR      RR        ....
EEEEEEEEEE      CCCCCCCC    HH      HH    RR      RR        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

```
.MAIN.  MACRO M1113  08-FEB-80 11:50
TABLE OF CONTENTS

     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    E$CHR            OBTAIN LINE OF INPUT FROM STRING
```



```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 1
```

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 1


      1        000000                 RT11    =  0          ;NOT RT-11
      2        000000                 RSTS    =  0          ;NOT RSTS/E
      3        000115                 RSX     = 'M          ;RSX-11/M
      4
      5        000000                 EIS     =  0          ;NO EIS
      6        000000                 FIS     =  0          ;NO FIS
      7        000000                 FPP     =  0          ;NO FPP
      8
      9        000004                 FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                 DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```



```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 2
```

L 14


```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

M 14

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  ECHR
                                        .IDENT  \V02.03\

                                        .SBTTL  E$CHR           OBTAIN LINE OF INPUT FROM STRING

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       THE INPUT STREAM IS SWITCHED TO THE E$CHR ROUTINE WHEN AN EXECUTE
                                        ;       OPERATION IS PERFORMED.  E$CHR, POINTED TO BY 'INPTR' PUTS THE NEXT
                                        ;       LINE IN THE STRING (DEFINED BY E$PT AND E$CT) INTO THE INPUT LINE
                                        ;       BUFFER DEFINED BY THE CALLER.  'F.END' IS SET IN THE FLAGS WORD IF
                                        ;       THE END OF THE INPUT STRING IS REACHED.
                                        ;
                                        ; INPUTS:
                                        ;       R1      - POINTS TO FIRST AVAILABLE BYTE OF  INPUT LINE BUFFER
                                        ;       R2      - LENGTH OF LINE BUFFER
                                        ;
                                        ; OUTPUTS:
                                        ;       R0      - POINTS TO START OF LINE BUFFER
                                        ;       R1      - BYTE COUNT OF CHARS PLACED INTO LINE BUFFER
                                        ;--
000000                                          .PSECT  CODE

000000                                  E$CHR::
000000                                          PUSH    R1                      ;SAVE POINTER TO BUFFER
000002  042737  002000  004764'                 BIC     #F.END,FLAGS            ;ASSUME NOT END OF STRING
000010  013703  005060'                         MOV     E$PT,R3                 ;POINT TO NEXT CHAR
000014  013704  005056'                         MOV     E$CT,R4                 ;COUNT OF CHARS REMAINING

000020  005704                          1$:     TST     R4                      ;END OF STRING?
000022  001007                                  BNE     2$                      ;NOPE
000024  052737  002000  004764'                 BIS     #F.END,FLAGS            ;YES, MARK AS LAST LINE IN STRING
000032  112721  000015                         MOVB    #CHR.CR,(R1)+           ;STORE CR AS MARKER
000036  005302                                  DEC     R2                      ;COUNT IT
000040  000420                                  BR      4$                      ;AND EXIT

000042  112300                          2$:     MOVB    (R3)+,R0                ;GET NEXT CHAR
000044  005304                                  DEC     R4                      ;COUNT IT
000046  110021                                  MOVB    R0,(R1)+                ;STORE IN LINE BUFFER
000050  005302                                  DEC     R2                      ;COUNT IT
000052  001001                                  BNE     5$                      ;STILL ROOM LEFT
000054  104541                                  LNLNG                           ;LINE TOO LONG
000056  022700  000015                  5$:     CMP     #CHR.CR,R0              ;END OF LINE?
000062  001356                                  BNE     1$                      ;NO, KEEP TRANSFERRING DATA TO LINE BUFFER

000064  122713  000012                  3$:     CMPB    #CHR.LF,@R3             ;NEXT CHAR LF AFTER THE CR?
000070  001004                                  BNE     4$                      ;NOPE
000072  005704                                  TST     R4                      ;END OF STRING?
000074  001402                                  BEQ     4$                      ;YES
000076  105723                                  TSTB    (R3)+                   ;NO SO SKIP OVER LF
000100  005304                                  DEC     R4                      ;BUT COUNT IT

000102  010437  005056'                  4$:     MOV     R4,E$CT                 ;RESET NEW VALUES
000106  001003                                  BNE     6$                      ;NOT END OF STRING
000110  052737  002000  004764'                 BIS     #F.END,FLAGS            ;MAKE SURE END FLAG IS SET
000116  010337  005060'                  6$:     MOV     R3,E$PT                 ;TO PICK UP REST OF STRING
000122                                          POP     R0                      ;RESTORE POINTER TO LINE START
```

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 44-1
E$CHR           OBTAIN LINE OF  INPUT FROM STRING

     58 000124  160001                          SUB     R0,R1           ;CALCULATE LENGTH OF LINE
     59 000126                                  RETURN                  ;AND RETURN
```



```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46
```

C 15

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46
E$CHR           OBTAIN LINE OF INPUT FROM STRING

      1         000001                          .END
```

D 15

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46-1
```

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140      D.DIV = 000000      ELESIZ  005132R     025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760      D.DMD = 000010      ENDCOR  000004R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762      D.DQAD= 177777      EOF    = 104475
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766      D.DROP= 000002      ERRMAX= 000166
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764      D.ENCD= 000002      ERRNUM  004772R     025
AC5    =%000005          CHRPDL= 000271    C.CZ   = 001000     D.EQ   = 000000     ERRPTR  004770R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG  = 000040     D.EXPD= 000002      ERR0    004774R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000      D.FGET= 000005      EXEBUF  005054R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010      D.FMT  = 000004     E$CHR   000000RG    020
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000      D.FPNT= 000005      E$CT    005056R     025
A.ARG  = 000040          CHRRBR= 000047    C.FRC  = 000001     D.FPUT= 000005      E$PT    005060R     025
A.ASB  = 000200          CHRRPR= 000042    C.IDEN= 000010      D.GE   = 000000     FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102      D.GT   = 000000     FA.CRE= 000010
A.CON  = 000020          CHRSMO= 000112    C.NOBF= 000400      D.IBM  = 000002     FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NUM  = 000020     D.IND  = 000002     FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.OP   = 000004     D.INNR= 000013      FA.EXC= 002000
A.NUM  = 000006          CHR.BS= 000010    C.OPN  = 010000     D.INTR= 000006      FA.EXT= 000004
A.PTR  = 000000          CHR.CC= 000003    C.RDLN= 177776      D.LE   = 000000     FA.NSP= 000100
A.TMP  = 000010          CHR.CR= 000015    C.REWT= 002000      D.LOG  = 000001     FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.RND  = 000010     D.LT   = 000000     FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756      D.MAX  = 000000     FA.RD  = 000001
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100      D.MEM  = 000002     FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200      D.MIN  = 000000     FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001      D.MUL  = 000000     FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002      D.NE   = 000000     FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SRC  = 177774     D.NN   = 000000     FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024      D.NR   = 000000     FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.TT   = 000004     D.OR   = 000000     FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770      D.PLUS= 000000      FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772      D.PSUB= 000006      FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    DEBUG  = 000000     D.PWR  = 000001     FD.CR  = 000002
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405      D.QQAD= 000000      FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R     025 D.QUAD= 000000  FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470      D.RES  = 000000     FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R     025 D.RHO  = 000000 FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIMAX  = 000020     D.ROT  = 000003     FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R     025 D.SCON= 000006  FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006  FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010          DIMTYP  003116R     025 D.SUB  = 000000 FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466          D.SUBS= 000006  FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417          D.TAKE= 000002  FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R     025 DPTERR= 104416          D.TRAN= 000004  FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425          DYDCNT  003224R     025 D.UNON= 000006  FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R     025 D.AND  = 000000         D.$MAX= 000000  FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.BXCT= 000007          D.$MIN= 000000  FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R     025 D.BXIO= 000007          D.$MUL= 000000  FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R     025 D.BXNL= 000007          D.$PLU= 000000  FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200          D.BXPP= 000007          D.$RES= 000000  FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010          D.BXPW= 000007          D.$SUB= 000000  FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020          D.BXRL= 000007          D.1COM= 000001  FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040          D.CIRC= 000001          D.1EXP= 000002  FF.NV  = 000003
CHRDLT= 000110           CS.ONE= 000100          D.COMB= 000001          D.1ROT= 000003  FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R     025 D.COMP= 000001          EDICHR  005063R     025 FF.RWD= 000001
CHREP  = 000105          CTMAX   000212R     025 D.CONT= 000006          EDICNT  005064R     025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R     025 D.CTLM= 000000          EDIFLG  005062R     025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162          D.DEAL= 000003          EDIT   = 000003         FHEADR  005012R     025
CHRFRO= 000135           C.BLOK= 000064          D.DECD= 000002          EIS    = 000000         FHRET   005040R     025
```



```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46-2
```

SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464          MONSFT  003126R  025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540          MONTYP  003124R  025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565          MONVAL  003134R  025 M.SIGN= 000000
FIXPTR  003074R  025 F.FUN = 000010          INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLAGS   004764R  025 F.FVER= 000120          INNTMP  003250R  025 M.BOL = 000200          M.SYM = 000203
FLCBUF  003250R  025 F.HEAD= 000006          INNTM2  005016R  025 M.BXAV= 000007          M.SYN = 000201
FLINE   005014R  025 F.HIBK= 000004          INPTR   000064R  025 M.BXCR= 000007          M.TRAN= 000012
FLOEXP  003270R  025 F.LBL = 000020          INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLONEG  003272R  025 F.LINE= 000012          IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLSWTB  000222R  025 F.LOC = 000004          I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FNINSI  005044R  025 F.LUN = 000042          I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FORIGN  000032R  025 F.MBCT= 000054          I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R  025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R  025 M.BXTS= 000007          NB.DEV= 000200
FPMODE  000042R  025 F.NREC= 000030          LARGBL  003162R  025 M.BXWA= 000007          NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R  025 M.CEI = 177777          NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R  025 M.CHR = 000202          NB.SD1= 000400
FREBLK  007400R  025 F.PREV= 000000          LARGLE  003174R  025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R  025 F.PRNT= 100000          LARGLN  003166R  025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R  025 F.PROT= 010000          LARGPT  003164R  025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R  025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R  025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R  025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R  025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R  025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R  025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R  025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R  025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R  025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R  025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R  025 M.INSP= 000006          NEGSGN  005063R  025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R  025 M.IOTA= 000000          NINPT1  003302R  025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020          M.LIT = 000205          NINPT2  003300R  025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R  025 M.LN  = 000001          NINSTK  003304R  025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541          M.MMD = 000010          NINTMP  003310R  025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R  025 M.NEG = 000000          NINTYP  003306R  025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R  025 M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R  025 M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R  025 M.NUM = 000204          NPRT1   004762R  025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012          NULL    007364R  025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R  025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R  025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R  025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R  025
F.ERR = 000052          GNOME   003102R  025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R  025
F.FACC= 000043          HASHC   005136R  025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R  025
F.FFBY= 000014          HASHTB  005140R  025 MONBLK  003120R  025 M.RDCT= 000011          NUMQFL  003234R  025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R  025 M.REVS= 000003          NUMRIG  003231R  025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R  025 M.ROLL= 000000          NUMWID  003226R  025
F.FNAM= 000110          ICHAN   005050R  025 MONPTR  003122R  025 M.SATA= 000001          NUMZFL  003235R  025
```

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46-3
SYMBOL TABLE

NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.CLR= 000040          T.LIT = 000020
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR = 000000          SY.ERP= 000020          T.MON = 000010
N.DESC= 000020          RARGLE  003216R     025 R.SEQ = 000003          SY.NOU= 000100          T.NIL = 000004
N.DEV = 000032          RARGLN  003210R     025 R.VAR = 000002          SY.OPN= 000002          T.NUM = 002000
N.DEVN= 000034          RARGPT  003206R     025 R6    =%000006          SY.PMT= 001000          T.OPFN= 000001
N.DID = 000024          RARGSF  003214R     025 R7    =%000007          SY.SRT= 000400          T.OTHR= 040000
N.DVNM= 000032          RARGTY  003212R     025 SI    =%000005          SY.SVC= 000004          T.SOP = 000002
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433          SY.SVL= 000200          T.SPOP= 000040
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R   025 S.AB  = 177776          T.TMP = 000200
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R   025 S.ASB = 100000          T.VAL = 100000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R   025 S.BFHD= 000020          T.VAR = 001000
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R   025 S.BITS= 000002          UPPER = 177400
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200          S.CHR1= 000003          UTNAME  000344R     023
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R   025 S.FATT= 000016          VALERR= 104413
N.PPN = 000024          REENTR  003060R     025 SI.BIT= 177774          S.FDB = 000140          VARBLE  007400R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772          S.FLB = 040000          VARBLS  000000R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.RET= 177772          S.FNAM= 000006          VERNUM  000000R     025
OBUFLN= 001000          RESERR= 104440          SI.VAL= 177776          S.FNB = 000036          VRSION= 000002
OCHAN   005052R   025 RESLEN  003150R     025 SKPSIZ  005134R   025 S.FNBW= 000017          WIDTH   000072R     025
OMODE   000402R   023 RESPTR  003140R     025 SPHOLD  000004R   024 S.FNTY= 000004          WROUND  000020R     025
OPTMP0  005066R   025 RESSFT  003146R     025 SPINIT  000002R   024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP1  005070R   025 RESTYP  003144R     025 STKINC= 000774          S.LEN = 000002          WSFULL= 104403
OPTMP2  005072R   025 RESULT  003136R     025 STOP  = 104420          S.LINK= 177774          WSNAME  000074R     025
OPTMP3  005074R   025 RESVAL  003154R     025 SUBCNT  003104R   025 S.NFEN= 000020          WSTIME  000132R     025
OPTMP4  005076R   025 RINC    005104R     025 SUBDCT  003106R   025 S.TYPE= 140000          $FIS  = ****** G
OPTMP5  005100R   025 RLTBAS  005066R     025 SUBPTR  003110R   025 S.VAL = 000000          $FPP  = ****** G
OPTMP6  005102R   025 ROTINC  005070R     025 SYMB1   005344R   025 TKDIMS  005072R   025 $FPPLN= ****** G
OPTMP7  005104R   025 ROUND   000010R     025 SYMF    005340R   025 TKFRAM  005066R   025 $MACHN= ****** G
OPTMP8  005106R   025 ROVER   003312R     025 SYMPT1  005346R   025 TKLEFT  005070R   025 $RSTS = ****** G
OPTMP9  005110R   025 RSTS  = 000000          SYMTAB  005352R   025 TMODE   000403R   023 $RSX  = ****** G
OPTM10  005112R   025 RSX   = 000115          SYMTBF= 104404          TMPTAG= 000166          $RT11 = ****** G
OPTM14  005026R   025 RTOP    005070R     025 SYMTOP  005350R   025 TTYBUF  004140R   025 .BOOLE= 000002
ORIGIN  000030R   025 RT11  = 000000          SYMTP   006272R   025 TTYLEN= 000622          .CHARA= 000000
OUTPTR  000066R   025 R.BITS= 000003          SYMWRN= 000054          TURKEY= 104436          .DEF. = 000002
OVROPR= 000000          R.FIX = 000001          SYNERR= 104407          TYPE  = 000166          .FLOAT= 000006
PATCH   007344R   025 R.LCT = 000002          SYSERR= 104402          T.BOL = 020000          .FSRCB= ****** G
PRECSN  000052R   025 R.LN  = 000002          SYSTAT  000000R   024 T.CON = 004000          .INTEG= 000004
RANDAC  003064R   025 R.LNUM= 000010          SY.BAD= 000001          T.DIM = 000100          .MOLUN  000404R     023
RANDY   003062R   025 R.NEXT= 000006          SY.BRK= 100000          T.FUN = 010000          ...GBL= 000000
RANERR= 104411          R.OFF = 000000          SY.BS = 000010          T.LIST= 000400          ...TPC= 000020
RARGBL  003204R   025

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

G 15

```
ECHR    MACRO M1113  08-FEB-80 11:50  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000130      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12918 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:47
ECHR,DM1:[305,301]ECHR/-SP=APLRX1,APLDEF,ECHR,END
```

H 15

```
**FILE**ID**EMPTY
```

```
**FILE**ID**EMPTY
```

```
EEEEEEEEEE   MM      MM   PPPPPPPP     TTTTTTTTTT   YY      YY
EEEEEEEEEE   MM      MM   PPPPPPPP     TTTTTTTTTT   YY      YY
EE           MMMM  MMMM   PP      PP       TT       YY      YY
EE           MMMM  MMMM   PP      PP       TT       YY      YY
EE           MM  MM  MM   PP      PP       TT         YY  YY
EE           MM  MM  MM   PP      PP       TT         YY  YY
EEEEEEEE     MM      MM   PPPPPPPP         TT           YY
EEEEEEEE     MM      MM   PPPPPPPP         TT           YY
EE           MM      MM   PP               TT           YY
EE           MM      MM   PP               TT           YY
EE           MM      MM   PP               TT           YY          ....
EE           MM      MM   PP               TT           YY          ....
EEEEEEEEEE   MM      MM   PP               TT           YY          ....
EEEEEEEEEE   MM      MM   PP               TT           YY          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```



```
.MAIN.  MACRO M1113  08-FEB-80 11:51
```

```
.MAIN.  MACRO M1113  08-FEB-80 11:51
TABLE OF CONTENTS

     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    EMPTY            CREATE AN EMPTY ARRAY
```



```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 1
```

```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 1


      1        000000                    RT11    =  0          ;NOT RT-11
      2        000000                    RSTS    =  0          ;NOT RSTS/E
      3        000115                    RSX     = 'M          ;RSX-11/M
      4
      5        000000                    EIS     =  0          ;NO EIS
      6        000000                    FIS     =  0          ;NO FIS
      7        000000                    FPP     =  0          ;NO FPP
      8
      9        000004                    FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                    DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```



```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 2
```

```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```



```
APL     MACRO M1113  08-FEB-80 11:51  PAGE 42
```

```
APL     MACRO M1113  08-FEB-80 11:51  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 44
```

```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  EMPTY
                                        .IDENT  \V02.03\

                                        .SBTTL  EMPTY           CREATE AN EMPTY ARRAY

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       'EMPTY' CREATES AN EMPTY ARRAY OF THE SAME DIMENSIONS AS THE
                                        ;       ONE POINTED TO BY R4.  AN ENTRY IS ALSO MADE ON THE SI FOR THE
                                        ;       ARRAY.
                                        ;
                                        ; INPUTS:
                                        ;       R4 -    POINTER TO SOURCE ARRAY
                                        ;--
                                                .PSECT  CODE

000000                                  EMPTY::
000000  116400  000002                          MOVB    A.DIMS(R4),R0   ;GET DIM COUNT
000004                                          CONINC  R0              ;;CHEAT FOR SCALAR
000010                                          PUSH    R0              ;SAVE FOR LATER
000012  006300                                  ASL     R0              ;MAKE INTO WORD COUNT
000014  062700  000006                          ADD     #A.DIM1,R0      ;PLUS HEADER SIZE
000020                                          PUSH    R0              ;SAVE OFFSET SIZE
000022                                          CALL    GETCOR          ;GET SOME CORE FOR THE RESULT
000026  012725  000000G                         MOV     #SYSER,(SI)+    ;NO RETURN FROM THIS FRAME
000032  012725  102000                          MOV     #T.NUM+T.VAL,(SI)+;SHOW IT'S A VALUE
000036  132764  000006  000003                  BITB    #A.NUM,A.BITS(R4);BUT IS IT NUMARIC?
000044  001003                                  BNE     24$             ;YES, SO LEAVE ALONE
000046  042765  002000  177776                  BIC     #T.NUM,-2(SI)   ;CLEAR NUMARIC FLAG FOR LITTERALS
000054  010025                          24$:    MOV     R0,(SI)+        ;POINT TO RESULT
000056  010010                                  MOV     R0,(R0)         ;POINT TO SELF
000060  062620                                  ADD     (SP)+,(R0)+     ;ADD OFFSET WORD
000062  012602                                  MOV     (SP)+,R2        ;SAVE # OF DIMS
000064  110220                                  MOVB    R2,(R0)+        ;FILL IN DIMS
000066                                          CONINC  R2              ;CHEAT FOR SCALAR
000072  116410  000003                          MOVB    A.BITS(R4),(R0) ;FILL IN BITS
000076  142720  177571                          BICB    #^C<A.NUM+A.ASB>,(R0)+ ;KILL EXCESS BITS
000102  005020                                  CLR     (R0)+           ;NO LENGTH
000104  005020                          23$:    CLR     (R0)+           ;CLEAR SIZE ALONG THIS DIM
000106                                          SOB     R2,23$          ;BE SURE TO DO ALL DIMS
000112                                          RETURN                  ;DONE!!!
```



```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 47
```

```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 47
EMPTY           CREATE AN EMPTY ARRAY

      1         000001                          .END
```

EMPTY "ACRO M1113 08-FEB-80 11:51 PAGE 47-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073          C.BLSZ= 000140          D.DIV = 000000          ELESIZ  005132R     025
AC1    =%000001          CHRLPR= 000072          C.BUFF= 177760          D.DMD = 000010          EMPTY   000000RG    020
AC2    =%000002          CHRMNS= 000137          C.BUFL= 177762          D.DQAD= 177777          ENDCOR  000004R     025
AC3    =%000003          CHRMUL= 000075          C.CCNT= 177766          D.DROP= 000002          EOF    = 104475
AC4    =%000004          CHRNEG= 000100          C.CPTR= 177764          D.ENCD= 000002          ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271          C.CZ  = 000100          D.EQ  = 000000          ERRNUM  004772R     025
ALLINT  003274R     025  CHRPLU= 000055          C.DIG = 000040          D.EXPD= 000002          ERRPTR  004770R     025
ARYRET  003276R     025  CHRPWR= 000120          C.FASW= 020000          D.FGET= 000005          ERRO    004774R     025
ASCQUO= 000042           CHRQST= 000121          C.FLEN= 000010          D.FMT = 000004          EXEBUF  005054R     025
A.ALEN= 000004           CHRQUO= 000113          C.FLSH= 004000          D.FPNT= 000005          E$CT    005056R     025
A.ARG = 000040           CHRRBR= 000047          C.FRC = 000001          D.FPUT= 000005          E$PT    005060R     025
A.ASB = 000200           CHRRPR= 000042          C.IDEN= 000010          D.GE  = 000000          FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074          C.NAME= 000102          D.GT  = 000000          FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112          C.NOBF= 000400          D.IBM = 000002          FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040          C.NUM = 000020          D.IND = 000002          FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131          C.OP  = 000004          D.INNR= 000013          FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010          C.OPN = 010000          D.INTR= 000006          FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003          C.RDLN= 177776          D.LE  = 000000          FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015          C.REWT= 002000          D.LOG = 000001          FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032          C.RND = 000010          D.LT  = 000000          FA.POS= 010000
BADERR= 104437           CHR.GT= 000046          C.SBIT= 177756          D.MAX = 000000          FA.RD = 000001
BLZBUB  005116R     025  CHR.LF= 000012          C.SPFX= 000100          D.MEM = 000002          FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043          C.SPGL= 000200          D.MIN = 000000          FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017          C.SPLX= 000001          D.MUL = 000000          FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057          C.SPRS= 000002          D.NE  = 000000          FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016          C.SRC = 177774          D.NN  = 000000          FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011          C.SRCP= 000024          D.NR  = 000000          FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106          C.TT  = 000004          D.OR  = 000000          FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141          C.WIDC= 177770          D.PLUS= 000000          FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145          C.WIDR= 177772          D.PSUB= 000006          FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163          DEBUG = 000000          D.PWR = 000001          FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164          DEFERR= 104405          D.QQAD= 000000          FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165          DEVIL   005114R     025 D.QUAD= 000000          FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172          DEVNTF= 104470          D.RES = 000000          FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060          DIGITS  000006R     025 D.RHO = 000000          FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061          DIMAX = 000020          D.ROT = 000003          FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001          DIMBLK  003112R     025 D.SCON= 000006          FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006          FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010          DIMTYP  003116R     025 D.SUB = 000000          FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466          D.SUBS= 000006          FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417          D.TAKE= 000002          FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R     025 DPTERR= 104416          D.TRAN= 000004          FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425          DYDCNT  003224R     025 D.UNON= 000006          FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R     025 D.AND = 000000          D.$MAX= 000000          FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R     025 D.BXCT= 000007          D.$MIN= 000000          FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R     025 D.BXIO= 000007          D.$MUL= 000000          FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R     025 D.BXNL= 000007          D.$PLU= 000000          FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200          D.BXPP= 000007          D.$RES= 000000          FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010          D.BXPW= 000007          D.$SUB= 000000          FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020          D.BXRL= 000007          D.1COM= 000001          FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040          D.CIRC= 000001          D.1EXP= 000002          FF.NV = 000003
CHRDLT= 000110           CS.ONE= 000100          D.COMB= 000001          D.1ROT= 000003          FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R     025 D.COMP= 000001          EDICHR  005063R     025 FF.RWD= 000001
CHREP = 000105           CTMAX   000212R     025 D.CONT= 000006          EDICNT  005064R     025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R     025 D.CTLM= 000000          EDIFLG  005062R     025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162          D.DEAL= 000003          EDIT   = 000003         FHEADR  005012R     025
CHRFRO= 000135           C.BLOK= 000064          D.DECD= 000002          EIS    = 000000         FHRET   005040R     025
```


EMPTY MACRO M1113 08-FEB-80 11:51 PAGE 47-2

SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001
FIS   = 000000          F.FNDF= 000010          ILFNAM= 104464      MONSFT  003126R 025 M.SCAN= 000011
FISCON= 046000          F.FTYP= 000116          ILLCHN= 104540      MONTYP  003124R 025 M.SHAP= 000000
FIXPTR  003074R 025     F.FUN = 000010          ILLSW = 104565      MONVAL  003134R 025 M.SIGN= 000000
FLAGS   004764R 025     F.FVER= 000120          INDERR= 104410      M.ABS = 000000      M.SUBR= 000001
FLCBUF  003250R 025     F.HEAD= 000006          INNTMP  003250R 025 M.BOL = 000200      M.SYM = 000203
FLINE   005014R 025     F.HIBK= 000004          INNTM2  005016R 025 M.BXAV= 000007      M.SYN = 000201
FLOEXP  003270R 025     F.LBL = 000020          INPTR   000064R 025 M.BXCR= 000007      M.TRAN= 000012
FLONEG  003272R 025     F.LINE= 000012          INSNAM= 104462      M.BXCT= 000007      M.$ABS= 000000
FLSWTB  000222R 025     F.LOC = 000004          IOERR = 104477      M.BXEX= 000007      M.$CEI= 000000
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776      M.BXFX= 000007      M.$FLR= 000000
FNINSI  005044R 025     F.LUN = 000042          I.DIM = 177772      M.BXIO= 000007      M.$IDN= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.LEN = 000012      M.BXLC= 000003      M.$NEG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062      M.BXNC= 000007      M.$SIG= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770      M.BXNL= 000007      M.1RDT= 000011
FO.RD = 000001          F.MON = 000002          I.PTR = 177774      M.BXPP= 000007      M.1REV= 000003
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766      M.BXPW= 000007      M.1SCN= 000011
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406      M.BXRL= 000007      NAMBUF  003240R 025
FPMODE  000042R 025     F.NREC= 000030          LABLPT  005042R 025 M.BXTS= 000007      NB.DEV= 000200
FPP   = 000000          F.OPN = 000010          LARGBL  003162R 025 M.BXWA= 000007      NB.DIR= 000100
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R 025 M.CEI = 177777      NB.NAM= 000004
FREBLK  007400R 025     F.PREV= 000000          LARGFL  003160R 025 M.CHR = 000202      NB.SD1= 000400
FUZZ    000142R 025     F.PRNT= 100000          LARGLE  003174R 025 M.CMT = 000206      NB.SD2= 001000
FXPTR1  003076R 025     F.PROT= 010000          LARGLN  003166R 025 M.DOUT= 000006      NB.SNM= 000040
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R 025 M.DQAD= 000006      NB.STP= 000020
F$EVI = 000200          F.RATT= 000001          LARGSF  003172R 025 M.ENCD= 000004      NB.SVR= 000010
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R 025 M.EXE = 000000      NB.TYP= 000002
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R 025 M.EXP = 000001      NB.VER= 000001
F.ACTL= 000076          F.READ= 004000          LBINC   005100R 025 M.FACT= 000001      ND.DEV= 000200
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R 025 M.FIL = 000177      ND.EXT= 000002
F.BBFS= 000062          F.RES = 000040          LBUFLN= 000621      M.FLR = 177777      ND.NAM= 000004
F.BDB = 000070          F.RET = 177772          LCOUNT  005106R 025 M.FMT = 000004      ND.PPN= 000100
F.BGBC= 000057          F.RSIZ= 000002          LENERR= 104412      M.GDUP= 000003      ND.$EX= 000000
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R 025 M.GDWN= 000003      ND.$ES= 000020
F.BKDN= 000026          F.R2  = 000014          LEXPTR  003100R 025 M.GETI= 000006      ND.$NM= 000000
F.BKDS= 000020          F.R3  = 000016          LFTOP   005066R 025 M.IBM = 000003      ND.$NS= 000040
F.BKEF= 000050          F.R4  = 000020          LINC    005102R 025 M.IDN = 000000      ND.$PN= 001400
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R 025 M.INSP= 000006      NEGSGN  005063R 025
F.BKST= 000024          F.SPDV= 000072          LININC  005026R 025 M.IOTA= 000000      NINPT1  003302R 025
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020      M.LIT = 000205      NINPT2  003300R 025
F.BR  = 000020          F.STBK= 000036          LITOP   005072R 025 M.LN  = 000001      NINSTK  003304R 025
F.CHR = 000075          F.STK = 000022          LNLNG = 104541      M.MMD = 000010      NINTMP  003310R 025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R 025 M.NEG = 000000      NINTYP  003306R 025
F.CODE= 000010          F.STP = 000001          LOWCOR  000002R 025 M.NOT = 000000      NOTAVL= 104472
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R 025 M.NUB = 000006      NOTOPN= 104473
F.DSPT= 000044          F.TRC = 000002          LWIDTH  000070R 025 M.NUM = 000204      NPRT1   004762R 025
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003      M.OUTR= 000012      NULL    007364R 025
F.EFBK= 000010          F.URBD= 000020          L.CNT = 000002      M.PITM= 000000      NUMCHN= 000020
F.EFN = 000050          F.VAL = 177776          L.PRNT= 000001      M.PRDS= 000006      NUMDFL  003230R 025
F.END = 002000          F.VBN = 000064          L.PTR = 000000      M.QQAD= 000000      NUMEFL  003233R 025
F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004      M.QUAD= 000000      NUMESN  003236R 025
F.ERR = 000052          GETCOR= ****** GX       L.SUBS= 000002      M.RAND= 000012      NUMFLG  003230R 025
F.FACC= 000043          GNOME   003102R 025     MACHIN= 000106      M.RAVL= 000000      NUMLEN  000062R 025
F.FFBY= 000014          HASHC   005136R 025     MAXCHN= 000014      M.RCIP= 000000      NUMNEG  003232R 025
F.FLAG= 000002          HASHTB  005140R 025     MONBLK  003120R 025 M.RDCT= 000011      NUMQFL  003234R 025
F.FLB = 000100          HSHMSK= 177601          MONCVT  003132R 025 M.REVS= 000003      NUMRIG  003231R 025
F.FNAM= 000110          HSHSIZ= 000200          MONLEN  003130R 025 M.ROLL= 000000      NUMWID  003226R 025
```

EMPTY MACRO M1113 08-FEB-80 11:51 PAGE 47-3
SYMBOL TABLE

```
NUMZFL  003235R      025 RARGBL  003204R      025 R.PREV= 0000      SY.CLR= 000040          T.LIT  = 000020
NYIERR= 104432           RARGCV  003220R      025 R.PTR  = 0000      SY.ERP= 000020          T.MON  = 000010
N.BLSZ= 000036           RARGFL  003202R      025 R.SEQ  = 000003    SY.NOU= 000100          T.NIL  = 000004
N.DESC= 000020           RARGLE  003216R      025 R.VAR  = 000002    SY.OPN= 000002          T.NUM  = 002000
N.DEV = 000032           RARGLN  003210R      025 R6     =%000006    SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034           RARGPT  003206R      025 R7     =%000007    SY.SRT= 000400          T.OTHR= 040000
N.DID = 000024           RARGSF  003214R      025 SI     =%000005    SY.SVC= 000004          T.SOP  = 000002
N.DVNM= 000032           RARGTY  003212R      025 SIDAMN= 104433     SY.SVL= 000200          T.SPOP= 000040
N.EXT = 000014           RARGVA  003222R      025 SIFRAM  003314R 025 S.AB  = 177776          T.TMP  = 000200
N.FID = 000000           RBASE   005074R      025 SISTAK  006350R 025 S.ASB = 100000          T.VAL  = 100000
N.FNAM= 000006           RCOUNR  005112R      025 SITOP   006346R 025 S.BFHD= 000020          T.VAR  = 001000
N.FTYP= 000014           RCOUNT  005110R      025 SITP    007144R 025 S.BITS= 000002          UPPER  = 177400
N.FVER= 000016           REDCNT  005070R      025 SIWARN= 000200     S.CHR1= 000003          UTNAME  000344R     023
N.NAME= 000006           REDIGN  005066R      025 SIZTAB  000044R 025 S.FATT= 000016          VALERR= 104413
N.NEXT= 000022           REDOFF  005072R      025 SI.BIT= 177774     S.FDB  = 000140          VARBLE  007400R     025
N.PPN  = 000024          REENTR  003060R      025 SI.LEN= 177772     S.FLB  = 040000          VARBLS  000000R     025
N.STAT= 000020           RESBIT  003142R      025 SI.RET= 177772     S.FNAM= 000006          VERNUM  000000R     025
N.UNIT= 000034           RESCVT  003152R      025 SI.VAL= 177776     S.FNB  = 000036          VRSION= 000002
OBUFLN= 001000           RESERR= 104440           SKPSIZ  005134R 025 S.FNBW= 000017          WIDTH   000072R     025
OCHAN   005052R      025 RESLEN  003150R      025 SPHOLD  000004R 024 S.FNTY= 000004          WROUND  000020R     025
OMODE   000402R      023 RESPTR  003140R      025 SPINIT  000002R 024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP0  005066R      025 RESSFT  003146R      025 STKINC= 000774     S.LEN  = 000002          WSFULL= 104403
OPTMP1  005070R      025 RESTYP  003144R      025 STOP   = 104420     S.LINK= 177774          WSNAME  000074R     025
OPTMP2  005072R      025 RESULT  003136R      025 SUBCNT  003104R 025 S.NFEN= 000020          WSTIME  000132R     025
OPTMP3  005074R      025 RESVAL  003154R      025 SUBDCT  003106R 025 S.TYPE= 140000          $FIS   = ****** G
OPTMP4  005076R      025 RINC    005104R      025 SUBPTR  003110R 025 S.VAL  = 000000          $FPP   = ****** G
OPTMP5  005100R      025 RLTBAS  005066R      025 SYMB1   005344R 025 TKDIMS  005072R  025 $FPPLN= ****** G
OPTMP6  005102R      025 ROTINC  005070R      025 SYMF    005340R 025 TKFRAM  005066R  025 $MACHN= ****** G
OPTMP7  005104R      025 ROUND   000010R      025 SYMPT1  005346R 025 TKLEFT  005070R  025 $RSTS  = ****** G
OPTMP8  005106R      025 ROVER   003312R      025 SYMTAB  005352R 025 TMODE   000403R  023 $RSX   = ****** G
OPTMP9  005110R      025 RSTS   = 000000           SYMTBF= 104404     TMPTAG= 000166          $RT11  = ****** G
OPTM10  005112R      025 RSX    = 000115           SYMTOP  005350R 025 TTYBUF  004140R  025 .BOOLE= 000002
OPTM14  005026R      025 RTOP     005070R      025 SYMTP   006272R 025  TYLEN= 000622          .CHARA= 000000
ORIGIN  000030R      025 RT11   = 000000           SYMWRN= 000054     TURKEY= 104436          .DEF.  = 000002
OUTPTR  000066R      025 R.BITS= 000003           SYNERR= 104407     TYPE   = 000166         .FLOAT= 000006
OVROPR= 000000               R.FIX  = 000001      SYSER  = ****** GX T.BOL  = 020000          .FSRCB= ****** G
PATCH   007344R      025 R.LCT  = 000002          SYSERR= 104402     T.CON  = 004000         .INTEG= 000004
PRECSN  000052R      025 R.LN   = 000002          SYSTAT   000000R 024 T.DIM  = 000100         .MOLUN  000404R     023
RANDAC  003064R      025 R.LNUM= 000010           SY.BAD= 000001     T.FUN  = 010000         ...GBL= 000000
RANDY   003062R      025 R.NEXT= 000006           SY.BRK= 100000     T.LIST= 000400          ...TPC= 000020
RANERR= 104411               R.OFF  = 000000      SY.BS  = 000010

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BYPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```


EMPTY MACRO M1113 08-FEB-80 11:51 PAGE 47-4

E 16

```
EMPTY   MACRO M1113  08-FEB-80 11:51  PAGE 47-4
SYMBOL TABLE

MNTRTL  000000     017
CODE    000114     020
PATCH   000000     021
$$FSR1  000000     022
DEF$0   000406     023
DEF$1   000006     024
DEF$2   007400     025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12921 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:04
EMPTY,DM1:[305,301]EMPTY/-SP=APLRX1,APLDEF,EMPTY,END
```

F 16

```
**FILE**ID**ENDL
```

```
**FILE**ID**ENDL
```

```
EEEEEEEEEE    NN       NN    DDDDDDDD      LL
EEEEEEEEEE    NN       NN    DDDDDDDD      LL
EE            NN       NN    DD      DD    LL
EE            NN       NN    DD      DD    LL
EE            NNNN     NN    DD      DD    LL
EE            NNNN     NN    DD      DD    LL
EEEEEEEE      NN  NN   NN    DD      DD    LL
EEEEEEEE      NN  NN   NN    DD      DD    LL
EE            NN    NNNN    DD      DD    LL
EE            NN    NNNN    DD      DD    LL
EE            NN       NN    DD      DD    LL                  ....
EE            NN       NN    DD      DD    LL                  ....
EEEEEEEEEE    NN       NN    DDDDDDDD      LLLLLLLLLL          ....
EEEEEEEEEE    NN       NN    DDDDDDDD      LLLLLLLLLL          ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                T
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```



```
.MAIN.  MACRO M1113  08-FEB-80 11:54
TABLE OF CONTENTS
```

.MAIN. MACRO M1113 08-FEB-80 11:54
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS     OPERATOR OVERLAYS
     5-  268    DEFINITIONS     REGISTERS
     6-  331    DEFINITIONS     MEMORY ORGANIZATION
     7-  377    DEFINITIONS     ORDERING PSECTS
     8-  426    DEFINITIONS     SI PARSE FRAMES
     9-  474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS     SI EXECUTION FRAMES
    11-  580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS     CHRTAB BITS
    12-  624    DEFINITIONS     SYSTEM STATUS BITS
    13-  640    DEFINITIONS     F-FRAMES
    14-  724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS     LIST STORAGE BLOCK
    18-  911    DEFINITIONS     INDEXING STACK FRAME
    19-  940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS     CHANNEL BLOCK
    22- 1085    DEFINITIONS             RSX DATA
    23- 1110    DEFINITIONS     NAME BLOCKS
    33- 1612    DEFINITIONS     MACROS
    35- 1905    DEFINITIONS     APL CHARACTERS
    36- 1967    DEFINITIONS     ERRORS
    44-    4    ENDL            CHECK SYNTAX AT EOL
```


ENDL   MACRO M1113 08-FEB-80 11:54 PAGE 1

ENDL MACRO M1113 08-FEB-80 11:54 PAGE 1

```
                        000000          RT11    =  0            ;NOT RT-11
                        000000          RSTS    =  0            ;NOT RSTS/E
                        000115          RSX     = 'M            ;RSX-11/M

                        000000          EIS     =  0            ;NO EIS
                        000000          FIS     =  0            ;NO FIS
                        000000          FPP     =  0            ;NO FPP

                        000004          FPPLEN  =  4.           ;SINGLE PRECISION

                        000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```


ENDL MACRO M1113 08-FEB-80 11:54 PAGE 2

```
ENDL    MACRO M1113  08-FEB-80 11:54  PAGE 2
APLDEF


                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```


APL MACRO M1113 08-FEB-80 11:54 PAGE 42

```
APL     MACRO M1113  08-FEB-80 11:54  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```



```
ENDL    MACRO M1113  08-FEB-80 11:54  PAGE 44
```

```
                                .TITLE  ENDL
                                .IDENT  \V02.03\

                                .SBTTL  ENDL            CHECK SYNTAX AT EOL

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       CHECK FOR END OF LINE
                                ;--
                                        .PSECT  CODE

000000                          ENDL::  CMPB    #CHR.CR,R2      ;IS CHAR A CR?
000000  122702  000015                  BEQ     4$              ;YES, RETURN
000004  001410
000006  122702  000040          1$:     CMPB    #CHRSPA,R2      ;SPACE?
000012  001401                          BEQ     3$              ;YES, IGNORE IT
000014  104425                  2$:     COMERR                  ;COMMAND ERROR

000016                          3$:     GETC    4$              ;GET A CHAR
000016  005303                          DEC R3                          ;ANY CHARACTERS LEFT?
000020  002402                          BLT 4$                  ;NO, BRANCH TO 4$
000022  112402                          MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
000024  000765                          BR      ENDL            ;AND CHECK IT

000026                          4$:     RETURN
```

```
ENDL    MACRO M1113  08-FEB-80 11:54  PAGE 46
ENDL            CHECK SYNTAX AT EOL

     1          000001                          .END
```



```
ENDL    MACRO M1113  08-FEB-80 11:54  PAGE 46-1
SYMBOL TABLE
```

ENDL MACRO M1113 08-FEB-80 11:54 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    ENDL    000000RG    020
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    EOF    = 104475
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.EQ  = 000000    ERRNUM  004772R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.EXPD= 000002    ERRPTR  004770R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    ERR0    004774R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FMT = 000004    EXEBUF  005054R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FPNT= 000005    E$CT    005056R     025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPUT= 000005    E$PT    005060R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.GE  = 000000    FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GT  = 000000    FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.IBM = 000002    FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IND = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.INNR= 000013    FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INTR= 000006    FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.LE  = 000000    FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LOG = 000001    FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LT  = 000000    FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.MAX = 000000    FA.RD = 000001
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MEM = 000002    FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MIN = 000000    FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MUL = 000000    FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.NE  = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.NN  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NR  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.OR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    DEBUG = 000000    D.PWR = 000001    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R 025 D.QUAD= 000000   FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.RES = 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R 025 D.RHO = 000000   FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIMAX = 000020    D.ROT = 000003    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R 025 D.SCON= 000006   FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R 025 DIMPTR  003114R 025 D.SDIF= 000006  FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010    DIMTYP  003116R 025 D.SUB = 000000   FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R 025 DIRFUL= 104466    D.SUBS= 000006    FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R 025 DOMERR= 104417    D.TAKE= 000002    FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R 025 DPTERR= 104416    D.TRAN= 000004    FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R 025 D.UNON= 000006    FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R 025 D.AND = 000000    D.$MAX= 000000    FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R 025 D.BXCT= 000007    D.$MIN= 000000    FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R 025 D.BXIO= 000007    D.$MUL= 000000    FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R 025 D.BXNL= 000007    D.$PLU= 000000    FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200      D.BXPP= 000007    D.$RES= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010      D.BXPW= 000007    D.$SUB= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020      D.BXRL= 000007    D.1COM= 000001    FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040      D.CIRC= 000001    D.1EXP= 000002    FF.NV = 000003
CHRDLT= 000110           CS.ONE= 000100      D.COMB= 000001    D.1ROT= 000003    FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R 025 D.COMP= 000001    EDICHR  005063R 025 FF.RWD= 000001
CHREP = 000105           CTMAX   000212R 025 D.CONT= 000006    EDICNT  005064R 025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R 025 D.CTLM= 000000    EDIFLG  005062R 025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162      D.DEAL= 000003    EDIT   = 000003     FHEADR  005012R     025
CHRFRO= 000135           C.BLOK= 000064      D.DECD= 000002    EIS    = 000000     FHRET   005040R     025
```

B	1		J	5		E	10		M	14	
C	1		K	5		F	10		N	14	
D	1		L	5		G	10		B	15	
E	1		M	5		H	10		C	15	
F	1		N	5		I	10		D	15	
G	1		B	6		J	10		E	15	
H	1	COPYC	C	6		K	10		F	15	
I	1		D	6		L	10		G	15	
J	1		E	6	DATA	M	10		H	15	EMPTY
K	1		F	6		N	10	DYAD	I	15	
L	1		G	6		B	11		J	15	
M	1		H	6		C	11		K	15	
N	1		I	6		D	11		L	15	
B	2		J	6		E	11		M	15	
C	2		K	6		F	11		N	15	
D	2		L	6		G	11		B	16	
E	2		M	6		H	11		C	16	
F	2		N	6		I	11		D	16	
G	2		B	7		J	11		E	16	
H	2		C	7	DATIME	K	11		F	16	ENDL
I	2		D	7		L	11		G	16	
J	2		E	7		M	11		H	16	
K	2		F	7		N	11		I	16	
L	2		G	7		B	12		J	16	
M	2	CPIARG	H	7		C	12		K	16	
N	2		I	7		D	12		L	16	
B	3		J	7		E	12	DYDEXP	M	16	
C	3		K	7		F	12				
D	3		L	7		G	12				
E	3		M	7		H	12				
F	3		N	7		I	12				
G	3		B	8		J	12				
H	3		C	8		K	12				
I	3		D	8	DECNAM	L	12				
J	3		E	8		M	12				
K	3		F	8		N	12				
L	3		G	8		B	13				
M	3		H	8		C	13				
N	3		I	8		D	13	DYDRT			
B	4	CPOARG	J	8		E	13				
C	4		K	8		F	13				
D	4		L	8		G	13				
E	4		M	8		H	13				
F	4		N	8		I	13				
G	4		B	9		J	13				
H	4		C	9		K	13				
I	4		D	9	DOT	L	13				
J	4		E	9		M	13				
K	4		F	9		N	13				
L	4		G	9		B	14				
M	4		H	9		C	14				
N	4		I	9		D	14				
B	5		J	9		E	14				
C	5		K	9		F	14				
D	5		L	9		G	14				
E	5		M	9		H	14				
F	5		N	9		I	14	ECHR			
G	5	CRTFIL	B	10		J	14				
H	5		C	10	DROPC	K	14				
I	5		D	10		L	14				

ENDL MACRO M1113 08-FEB-80 11:54 PAGE 46-2
SYMBOL TABLE

```
FILNTF= 104467           F.FNB = 000102           ILFNAM= 104464           MONSFT  003126R  025 M.SCAN= 000011
FIS   = 000000           F.FNDF= 000010           ILLCHN= 104540           MONTYP  003124R  025 M.SHAP= 000000
FISCON= 046000           F.FTYP= 000116           ILLSW = 104565           MONVAL  003134R  025 M.SIGN= 000000
FIXPTR  003074R  025     F.FUN = 000010           INDERR= 104410           M.ABS = 000000       M.SUBR= 000001
FLAGS   004764R  025     F.FVER= 000120           INNTMP  003250R  025     M.BCL = 000200       M.SYM = 000203
FLCBUF  003250R  025     F.HEAD= 000006           INNTM2  005016R  025     M.BXAV= 000007       M.SYN = 000201
FLINE   005014R  025     F.HIBK= 000004           INPTR   000064R  025     M.BXCR= 000007       M.TRAN= 000012
FLOEXP  003270R  025     F.LBL = 000020           INSNAM= 104462           M.BXCT= 000007       M.$ABS= 000000
FLONEG  003272R  025     F.LINE= 000012           IOERR = 104477           M.BXEX= 000007       M.$CEI= 000000
FLSWTB  000222R  025     F.LOC = 000004           I.CNT = 177776           M.BXFX= 000007       M.$FLR= 000000
FLTLEN= 000010           F.LSTC= 000004           I.DIM = 177772           M.BXIO= 000007       M.$IDN= 000000
FNINSI  005044R  025     F.LUN = 000042           I.LEN = 000012           M.BXLC= 000003       M.$NEG= 000000
FORIGN  000032R  025     F.MBCT= 000054           I.OERR= 000062           M.BXNC= 000007       M.$SIG= 000000
FO.APD= 000106           F.MBC1= 000055           I.OLD = 177770           M.BXNL= 000007       M.1RDT= 000011
FO.MFY= 000002           F.MBFG= 000056           I.PTR = 177774           M.BXPP= 000007       M.1REV= 000003
FO.RD = 000001           F.MON = 000002           I.STR = 177766           M.BXPW= 000007       M.1SCN= 000011
FO.UPD= 000006           F.NIL = 000001           LABERR= 104406           M.BXRL= 000007       NAMBUF  003240R  025
FO.WRT= 000016           F.NPBD= 000024           LABLPT  005042R  025     M.BXTS= 000007       NB.DEV= 000200
FPMODE  000042R  025     F.NREC= 000030           LARGBL  003162R  025     M.BXWA= 000007       NB.DIR= 000100
FPP   = 000000           F.OPN = 000010           LARGCV  003176R  025     M.CEI = 177777       NB.NAM= 000004
FPPLEN= 000004           F.OVBS= 000030           LARGFL  003160R  025     M.CHR = 000202       NB.SD1= 000400
FREBLK  007400R  025     F.PREV= 000000           LARGLE  003174R  025     M.CMT = 000206       NB.SD2= 001000
FUZZ    000142R  025     F.PRNT= 100000           LARGLN  003166R  025     M.DOUT= 000006       NB.SNM= 000040
FXPTR1  003076R  025     F.PROT= 010000           LARGPT  003164R  025     M.DQAD= 000006       NB.STP= 000020
F$EVAL= 000100           F.RACC= 000016           LARGSF  003172R  025     M.ENCD= 000004       NB.SVR= 000010
F$EVI = 000200           F.RATT= 000001           LARGTY  003170R  025     M.EXE = 000000       NB.TYP= 000002
F$EXEC= 000400           F.RCNM= 000034           LARGVA  003200R  025     M.EXP = 000001       NB.VER= 000001
F$FUNC= 001000           F.RCTL= 000017           LBINC   005100R  025     M.FACT= 000001       ND.DEV= 000200
F.ACTL= 000076           F.READ= 004000           LBUFF   003316R  025     M.FIL = 000177       ND.EXT= 000002
F.ALOC= 000040           F.REDF= 000040           LBUFLN= 000621           M.FLR = 177777       ND.NAM= 000004
F.BBFS= 000062           F.RES = 000040           LCOUNT  005106R  025     M.FMT = 000004       ND.PPN= 000100
F.BDB = 000070           F.RET = 177772           LENERR= 104412           M.GDUP= 000003       ND.$EX= 000000
F.BGBC= 000057           F.RSIZ= 000002           LENTAB  000054R  025     M.GDWN= 000003       ND.$ES= 000020
F.BITS= 177774           F.RTYP= 000000           LEXPTR  003100R  025     M.GETI= 000006       ND.$NM= 000000
F.BKDN= 000026           F.R2  = 000014           LFTOP   005066R  025     M.IBM = 000003       ND.$NS= 000040
F.BKDS= 000020           F.R3  = 000016           LINC    005102R  025     M.IDN = 000000       ND.$PN= 001400
F.BKEF= 000050           F.R4  = 000020           LINE    005016R  025     M.INSP= 000006       NEGSGN  005063R  025
F.BKP1= 000051           F.SEQN= 000100           LININC  005026R  025     M.IOTA= 000000       NINPT1  003302R  025
F.BKST= 000024           F.SPDV= 000072           LISIZE= 000020           M.LIT = 000205       NINPT2  003300R  025
F.BKVB= 000064           F.SPUN= 000074           LITOP   005072R  025     M.LN  = 000001       NINSTK  003304R  025
F.BF  = 000020           F.STBK= 000036           LNLNG = 104541           M.MMD = 000010       NINTMP  003310R  025
F.CHR = 000075           F.SYK = 000022           LOCLCT  005036R  025     M.NEG = 000000       NINTYP  003306R  025
F.CNTG= 000034           F.SIOP= 000004           LOWCOR  000002R  025     M.NOT = 000000       NOTAVL= 104472
F.CODE= 000010           F.STP = 000001           LSTERP  004766R  025     M.NUB = 000006       NOTOPN= 104473
F.DFNB= 000046           F.SUSP= 000002           LWIDTH  000070R  025     M.NUM = 000204       NPRT1   004762R  025
F.DSPT= 000044           F.TRC = 000002           L.BITS= 000003           M.OUTR= 000012       NULL    007364R  025
F.DVNM= 000134           F.UNIT= 000136           L.CNT = 000002           M.PITM= 000000       NUMCHN= 000020
F.EFBK= 000010           F.URBD= 000020           L.PRNT= 000001           M.PRDS= 000006       NUMDFL  003230R  025
F.EFN = 000050           F.VAL = 177776           L.PTR = 000000           M.QQAD= 000000       NUMEFL  003233R  025
F.END = 002000           F.VBN = 000064           L.PTR1= 000004           M.QUAD= 000000       NUMESN  003236R  025
F.EOBB= 000032           F.VBSZ= 000060           L.SUBS= 000002           M.RAND= 000012       NUMFLG  003230R  025
F.ERR = 000052           GNOME   003102R  025     MACHIN= 000106           M.RAVL= 000000       NUMLEN  000062R  025
F.FACC= 000043           HASHC   005136R  025     MAXCHN= 000014           M.RCIP= 000000       NUMNEG  003232R  025
F.FFBY= 000014           HASHTB  005140R  025     MONBLK  003120R  025     M.RDCT= 000011       NUMQFL  003234R  025
F.FLAG= 000002           HSHMSK= 177601           MONCVT  003132R  025     M.REVS= 000003       NUMRIG  003231R  025
F.FLB = 000100           HSHSIZ= 000200           MONLEN  003130R  025     M.ROLL= 000000       NUMWID  003226R  025
F.FNAM= 000110           ICHAN   005050R  025     MONPTR  003122R  025     M.SATA= 000001       NUMZFL  003235R  025
```

```
ENDL    MACRO M1113  08-FEB-30 11:54  PAGE 46-3
SYMBOL TABLE

NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.CLR= 000040          T.LIT = 000020
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR = 000000          SY.ERP= 000020          T.MON = 000010
N.DESC= 000020          RARGLE  003216R     025 R.SEQ = 000003          SY.NOU= 000100          T.NIL = 000004
N.DEV = 000032          RARGLN  003210R     025 R.VAR = 000002          SY.OPN= 000002          T.NUM = 002000
N.DEVN= 000034          RARGPT  003206R     025 R6    =%000006          SY.PMT= 001000          T.OPFN= 000001
N.DID = 000024          RARGSF  003214R     025 R7    =%000007          SY.SRT= 000400          T.OTHR= 040000
N.DVNM= 000032          RARGTY  003212R     025 SI    =%000005          SY.SVC= 000004          T.SOP = 000002
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433          SY.SVL= 000200          T.SPOP= 000040
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R   025 S.AB  = 177776          T.TMP = 000200
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R   025 S.ASB = 100000          T.VAL = 100000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R   025 S.BFHD= 000020          T.VAR = 001000
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R   025 S.BITS= 000002          UPPER = 177400
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200          S.CHR1= 000003          UTNAME  000344R     023
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R   025 S.FATT= 000016          VALERR= 104413
N.PPN = 000024          REENTR  003060R     025 SI.BIT= 177774          S.FDB = 000140          VARBLE  007400R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772          S.FLB = 040000          VARBLS  000000R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.RET= 177772          S.FNAM= 000006          VERNUM  000000R     025
OBUFLN= 001000          RESERR= 104440          SI.VAL= 177776          S.FNB = 000036          VRSION= 000002
OCHAN   005052R   025   RESLEN  003150R     025 SKPSIZ  005134R   025 S.FNBW= 000017          WIDTH   000072R     025
OMODE   000402R   023   RESPTR  003140R     025 SPHOLD  000004R   024 S.FNTY= 000004          WROUND  000020R     025
OPTMP0  005066R   025   RESSFT  003146R     025 SPINIT  000002R   024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP1  005070R   025   RESTYP  003144R     025 STKINC= 000774          S.LEN = 000002          WSFULL= 104403
OPTMP2  005072R   025   RESULT  003136R     025 STOP  = 104420          S.LINK= 177774          WSNAME  000074R     025
OPTMP3  005074R   025   RESVAL  003154R     025 SUBCNT  003104R   025 S.NFEN= 000020          WSTIME  000132R     025
OPTMP4  005076R   025   RINC    005104R     025 SUBDCT  003106R   025 S.TYPE= 140000          $FIS  = ****** G
OPTMP5  005100R   025   RLTBAS  005066R     025 SUBPTR  003110R   025 S.VAL = 000000          $FPP  = ****** G
OPTMP6  005102R   025   ROTINC  005070R     025 SYMB1   005344R   025 TKDIMS  005072R   025 $FPPLN= ****** G
OPTMP7  005104R   025   ROUND   000010R     025 SYMF    005340R   025 TKFRAM  005066R   025 $MACHN= ****** G
OPTMP8  005106R   025   ROVER   003312R     025 SYMPT1  005346R   025 TKLEFT  005070R   025 $RSTS = ****** G
OPTMP9  005110R   025   RSTS  = 000000          SYMTAB  005352R   025 TMODE   000403R   023 $RSX  = ****** G
OPTM10  005112R   025   RSX   = 000115          SYMTBF= 104404          TMPTAG= 000166          $RT11 = ****** G
OPTM14  005026R   025   RTOP    005070R     025 SYMTOP  005350R   025 TTYBUF  004140R   025 .BOOLE= 000002
ORIGIN  000030R   025   RT11  = 000000          SYMTP   006272R   025 TTYLEN= 000622          .CHARA= 000000
OUTPTR  000066R   025   R.BITS= 000003          SYMWRN= 000054          TURKEY= 104436          .DEF. = 000002
OVROPR= 000000          R.FIX = 000001          SYNERR= 104407          TYPE  = 000166          .FLOAT= 000006
PATCH   007344R   025   R.LCT = 000002          SYSERR= 104402          T.BOL = 020000          .FSRCB= ****** G
PRECSN  000052R   025   R.LN  = 000002          SYSTAT  000000R   024 T.CON = 004000          .INTEG= 000004
RANDAC  003064R   025   R.LNUM= 000010          SY.BAD= 000001          T.DIM = 000100          .MOLUN  000404R     023
RANDY   003062R   025   R.NEXT= 000006          SY.BRK= 100000          T.FUN = 010000          ...GBL= 000000
RANERR= 104411          R.OFF = 000000          SY.BS = 000010          T.LIST= 000400          ...TPC= 000020
RARGBL  003204R   025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

D 1

```
ENDL    MACRO M1113  08-FEB-80 11:54  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000030      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12909 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:40
ENDL,DM1:[305,301]ENDL/-SP=APLRX1,APLDEF,ENDL,END
```

F 1

E 1

FILEID**ERASE

```
EEEEEEEEEE   RRRRRRRR      AAAAAA      SSSSSSSS   EEEEEEEEEE
EEEEEEEEEE   RRRRRRRR      AAAAAA      SSSSSSSS   EEEEEEEEEE
EE           RR      RR   AA      AA   SS         EE
EE           RR      RR   AA      AA   SS         EE
EE           RR      RR   AA      AA   SS         EE
EE           RR      RR   AA      AA   SS         EE
EEEEEEEE     RRRRRRRR     AA      AA     SSSSSS   EEEEEEEE
EEEEEEEE     RRRRRRRR     AA      AA     SSSSSS   EEEEEEEE
EE           RR   RR      AAAAAAAAAA           SS  EE
EE           RR   RR      AAAAAAAAAA           SS  EE
EE           RR    RR     AA      AA           SS  EE                ....
EE           RR    RR     AA      AA           SS  EE                ....
EEEEEEEEEE   RR      RR   AA      AA   SSSSSSSS   EEEEEEEEEE        ....
EEEEEEEEEE   RR      RR   AA      AA   SSSSSSSS   EEEEEEEEEE        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

E 1

.MAIN. MACRO M1113 08-FEB-80 11:55
TABLE OF CONTENTS


```
      2-     1    APLDEF
      2-     8    A P L
      2-     9
      2-    10    A PROGRAMMING LANGUAGE
      2-    11
      2-    12    STEVE POULSEN & DAVE VANN
      2-    13    11-Jul-76
      2-    14    ROBERT GOTTLIEB & PETER HAYNES
      2-    15    2-Feb-79
      2-    16
      2-    17    V2
      2-    18
      2-    19
      2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
      2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
      2-    22
      2-    23
      2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
      2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
      2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
      2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
      2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
      2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
      2-    30    DEC.
      2-    31
      2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
      2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
      2-    34    CORPORATION.
      2-    35
      2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
      2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
      2-    38
      2-    39
      3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
      4-   139    DEFINITIONS      OPERATOR OVERLAYS
      5-   268    DEFINITIONS      REGISTERS
      6-   331    DEFINITIONS      MEMORY ORGANIZATION
      7-   377    DEFINITIONS      ORDERING PSECTS
      8-   426    DEFINITIONS      SI PARSE FRAMES
      9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
     10-   515    DEFINITIONS      SI EXECUTION FRAMES
     11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
     12-   607    DEFINITIONS      CHRTAB BITS
     12-   624    DEFINITIONS      SYSTEM STATUS BITS
     13-   640    DEFINITIONS      F-FRAMES
     14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
     16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
     17-   869    DEFINITIONS      LIST STORAGE BLOCK
     18-   911    DEFINITIONS      INDEXING STACK FRAME
     19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
     20-   969    DEFINITIONS      CHANNEL BLOCK
     22-  1085    DEFINITIONS                RSX DATA
     23-  1110    DEFINITIONS      NAME BLOCKS
     33-  1613    DEFINITIONS      MACROS
     35-  1905    DEFINITIONS      APL CHARACTERS
     36-  1967    DEFINITIONS      ERRORS
     44-     4    ERASE            ROUTINE TO ERASE ITEMS
```

```
                        
     1          000000          RT11    =  0          ;NOT RT-11
     2          000000          RSTS    =  0          ;NOT RSTS/E
     3          000115          RSX     = 'M          ;RSX-11/M
     4
     5          000000          EIS     =  0          ;NO EIS
     6          000000          FIS     =  0          ;NO FIS
     7          000000          FPP     =  0          ;NO FPP
     8
     9          000004          FPPLEN  =  4.         ;SINGLE PRECISION
    10
    11          000000          DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
ERASE   MACRO M1113  08-FEB-80 11:55  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 11:55  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  ERASE
                                .IDENT  \V02.03\

                                .SBTTL  ERASE           ROUTINE TO ERASE ITEMS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'ERASE' IS USED TO WIPE OUT VARIABLES AND FUNCTIONS.
                                ;       VARIABLES ARE EASY: SIMPLY RELEASE THEIR VALUES AND
                                ;       ZERO THE VALUE POINTER IN THE NAME BLOCK.  FUNCTIONS
                                ;       ARE HARDER.  FIRST OF ALL, A FUNCTION CANNOT BE ERASED
                                ;       IF IT IS IN THE SI STACK.  VERY EVIL THINGS HAPPEN
                                ;       WHEN THE FUNCTION TRIES TO CONTINUE EXECUTION AND FINDS
                                ;       ITSELF GONE!  EACH LINE OF A FUNCTION MUST BE DELETED
                                ;       INDIVIDUALLY, ALONG WITH THE LIST BLOCK CONTAINING THE
                                ;       POINTERS TO LINES WITH LABELS.  NOTE:  A SYMBOL BLOCK
                                ;       IS NEVER DESTROYED, ITS VALUE POINTER IS JUST SET TO
                                ;       ZERO.  THIS IS BECAUSE THE VARIABLE MAY BE LOCAL IN
                                ;       SOME FUNCTION IN THE SI STACK OR IT MAY HAVE A
                                ;       REFERENCE IN A FUNCTION.
                                ;
                                ; INPUTS:
                                ;       R0      - A pointer to where the pointer to the
                                ;                 global value was located.
                                ;
                                ; OUTPUTS:
                                ;       R0      - A pointer to where the pointer to the
                                ;                 global value was located. (i.e., the location
                                ;                 @R0 should be 0 if the erase was successful)
                                ;--
 000000                                 .PSECT  CODE

 000000                         ERASE:: PUSH    <R2,R1,R0>      ;Save some registers
 000006  011000                         MOV     (R0),R0         ;AND POINT TO DATA
 000010  001464                         BEQ     7$              ;Nothing to erase, finished
 000012  005760  000002                 TST     2.(R0)          ;Is it an Array?
 000016  002005                         BGE     1$              ;No, is something else
 000020  132760  000020  000003        BITB    #A.CON,A.BITS(R0);LABEL?
 000026  001055                         BNE     7$              ;YES, CAN'T ERASE LABELS!!!
 000030  000450                         BR      6$              ;Otherwise, is OK
                                ;+
                                ; ERASE A FUNCTION.  FIRST CHECK IF IT IS IN THE
                                ; SI STACK.
                                ;-
 000032  032760  040000  000002 1$:     BIT     #S.FLB,2.(R0)   ;Is it a function?
 000040  001444                         BEQ     6$              ;No, must be list
 000042  013701  003314'                MOV     SIFRAM,R1       ;POINT TO CURRENT FRAME

 000046  032761  001000  000002 2$:     BIT     #F$FUNC,F.FLAG(R1);F-FRAME?
 000054  001404                         BEQ     3$              ;NO
 000056  027661  000000  000010        CMP     @0(SP),F.FUN(R1);IS FN IN SI?
 000064  001436                         BEQ     7$              ;Yes, failure

 000066  011101                 3$:     MOV     (R1),R1         ;POINT TO PREVIOUS FRAME
 000070  001366                         BNE     2$              ;IF NOT LAST KEEP LOOKING
                                ;+
                                ; DELETE THE LIST BLOCK THEN EACH OF THE FUNCTION'S LINES
```

ERASE ROUTINE TO ERASE ITEMS

```
                                        ;-
000072  017601  000000                   MOV     @(SP),R1        ;POINT TO FN HEADER
000076  062701  000010                   ADD     #R.LNUM,R1      ;POINT TO LINE #
000102  063701  000062'                  ADD     NUMLEN,R1       ;POINT TO LIST PTR
000106  011100                           MOV     (R1),R0         ;GET LIST BLOCK POINTER
000110  001403                           BEQ     4$              ;NO LABELS
000112                                   CALL    RETCOR          ;KILL LIST
000116  005011                           CLR     @R1             ;NO MORE LABELS

000120  017601  000000           4$:     MOV     @(SP),R1        ;COPY PTR TO FN HEADER
000124  010161  000006                   MOV     R1,R.NEXT(R1)   ;FUNCTION HAS NO LINES NOW

000130  016101  000004           5$:     MOV     R.PREV(R1),R1   ;POINT TO LAST LINE
000134  010100                           MOV     R1,R0           ;COPY PTR TO LINE
000136  020176  000000                   CMP     R1,@(SP)        ;DONE?
000142  001403                           BEQ     6$              ;Yes, only header remains
000144                                   CALL    RETCOR          ;KILL THE LINE
000150  000767                           BR      5$              ;Loop for each line
                                        ;+
                                        ; Here, we clear the pointer to the <Whatever> and
                                        ; erase the final segment. We're done!
                                        ;-
000152  005076  000000           6$:     CLR     @(SP)           ;Undefine variable
000156                                   CALL    RETCOR          ;RELEASE VALUE

000162                           7$:     POP     <R0,R1,R2>      ;Restore the registers
000170  005710                           TST     (R0)            ;Set condition codes
000172                                   RETURN                  ;Finished
```

```
ERASE   MACRO M1113  08-FEB-80 11:55  PAGE 46
ERASE           ROUTINE TO ERASE ITEMS

      1         000001                          .END
```

ERASE MACRO M1113 08-FEB-80 11:55 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R     025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    EOF     = 104475
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    ERASE   000000RG    020
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.EQ  = 000000    ERRNUM  004772R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.EXPD= 000002    ERRPTR  004770R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    ERRO    004774R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FMT = 000004    EXEBUF  005054R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FPNT= 000005    E$CT    005056R     025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPUT= 000005    E$PT    005060R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.GE  = 000000    FA.APD= 000100
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GT  = 000000    FA.CRE= 000010
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.IBM = 000002    FA.DLK= 001000
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IND = 000002    FA.ENB= 100000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.INNR= 000013    FA.EXC= 002000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INTR= 000006    FA.EXT= 000004
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.LE  = 000000    FA.NSP= 000100
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LOG = 000001    FA.NSR= 000200
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LT  = 000000    FA.POS= 010000
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.MAX = 000000    FA.RD = 000001
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MEM = 000002    FA.RWD= 004000
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MIN = 000000    FA.SEQ= 040000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MUL = 000000    FA.SHR= 000040
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.NE  = 000000    FA.TMP= 000020
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.NN  = 000000    FA.WCK= 020000
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NR  = 000000    FA.WRT= 000002
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.OR  = 000000    FD.BLK= 000010
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FD.CCL= 000002
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FD.COM= 020000
CHAN05  001240R     025  CHR..S= 000163    DEBUG = 000000    D.PWR = 000001    FD.CR = 000002
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FD.DIR= 000010
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R   025 D.QUAD= 000000    FD.FTN= 000001
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.RES = 000000    FD.F11= 040000
CHAN09  002150R     025  CHR..0= 000060    DIGITS  000006R   025 D.RHO = 000000    FD.INS= 000010
CHAN10  002332R     025  CHR..1= 000061    DIMAX = 000020    D.ROT = 000003    FD.ISP= 002000
CHAN11  002514R     025  CH.AND= 000001    DIMBLK  003112R   025 D.SCON= 000006    FD.MNT= 100000
CHAN12  002676R     025  CMDBUF  003250R   025 DIMPTR  003114R   025 D.SDIF= 000006    FD.OSP= 004000
CHAN13  000022R     023  CMDMAX= 000010    DIMTYP  003116R   025 D.SUB = 000000    FD.PLC= 000004
CHAN14  000204R     023  CNTSIZ  005130R   025 DIRFUL= 104466    D.SUBS= 000006    FD.PRN= 000004
CHNOPN= 104471           COMBIN  005076R   025 DOMERR= 104417    D.TAKE= 000002    FD.PSE= 010000
CHRAND= 000051           COMCNT  005126R   025 DPTERR= 104416    D.TRAN= 000004    FD.RAH= 000001
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R   025 D.UNON= 000006    FD.RAN= 000002
CHRBKR= 000133           COMPTR  005122R   025 D.AND = 000000    D.$MAX= 000000    FD.REC= 000001
CHRBOL= 000200           COMRTN  003156R   025 D.BXCT= 000007    D.$MIN= 000000    FD.RWM= 000001
CHRBOX= 000114           COMSKP  005124R   025 D.BXIO= 000007    D.$MUL= 000000    FD.SDI= 000020
CHRCMA= 000054           CORPTR  005120R   025 D.BXNL= 000007    D.$PLU= 000000    FD.SQD= 000040
CHRCOL= 000076           CS.ASS= 000200        D.BXPP= 000007    D.$RES= 000000    FD.TTY= 000004
CHRDEL= 000107           CS.BOO= 000010        D.BXPW= 000007    D.$SUB= 000000    FD.WBH= 000002
CHRDIV= 000053           CS.FLO= 000020        D.BXRL= 000007    D.1COM= 000001    FF.CHR= 000005
CHRDLR= 000176           CS.IDN= 000040        D.CIRC= 000001    D.1EXP= 000002    FF.NV = 000003
CHRDLT= 000110           CS.ONE= 000100        D.COMB= 000001    D.1ROT= 000003    FF.POE= 000002
CHRDOT= 000056           CTEPS   000152R   025 D.COMP= 000001    EDICHR  005063R   025 FF.RWD= 000001
CHREP = 000105           CTMAX   000212R   025 D.CONT= 000006    EDICNT  005064R   025 FF.RWF= 000006
CHRERR= 104431           CTRMAX  000172R   025 D.CTLM= 000000    EDIFLG  005062R   025 FF.SPC= 000004
CHRFIL= 000177           C.ALOC= 000162        D.DEAL= 000003    EDIT  = 000003        FHEADR  005012R     025
CHRFRO= 000135           C.BLOK= 000064        D.DECD= 000002    EIS   = 000000        FHRET   005040R     025
```

SYMBOL TABLE

```
FILNTF= 104467          F.FNB  = 000102        ILFNAM= 104464          MONSFT  003126R 025 M.SCAN= 000011
FIS    = 000000          F.FNDF= 000010        ILLCHN= 104540          MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116        ILLSW = 104565          MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025 F.FUN = 000010        INDERR= 104410          M.ABS  = 000000         M.SUBR= 000001
FLAGS   004764R 025 F.FVER= 000120        INNTMP  003250R 025 M.BOL  = 000200         M.SYM  = 000203
FLCBUF  003250R 025 F.HEAD= 000006        INNTM2  005016R 025 M.BXAV= 000007         M.SYN  = 000201
FLINE   005014R 025 F.HIBK= 000004        INPTR   000064R 025 M.BXCR= 000007         M.TRAN= 000012
FLOEXP  003270R 025 F.LBL  = 000020        INSNAM= 104462          M.BXCT= 000007         M.$ABS= 000000
FLONEG  003272R 025 F.LINE= 000012        IOERR = 104477          M.BXEX= 000007         M.$CEI= 000000
FLSWTB  000222R 025 F.LOC  = 000004        I.CNT  = 177776          M.BXFX= 000007         M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004        I.DIM  = 177772          M.BXIO= 000007         M.$IDN= 000000
FNINSI  005044R 025 F.LUN  = 000042        I.LEN  = 000012          M.BXLC= 000003         M.$NEG= 000000
FORIGN  000032R 025 F.MBCT= 000054        I.OERR= 000062          M.BXNC= 000007         M.$SIG= 000000
FO.APD= 000100          F.MBC1= 000055        I.OLD  = 177770          M.BXNL= 000007         M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056        I.PTR  = 177774          M.BXPP= 000007         M.1REV= 000003
FO.RD  = 000001          F.MON  = 000002        I.STR  = 177766          M.BXPW= 000007         M.1SCN= 000011
FO.UPD= 000006          F.NIL  = 000001        LABERR= 104406          M.BXRL= 000007         NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024        LABLPT  005042R 025 M.BXTS= 000007         NB.DEV= 000200
FPMODE  000042R 025 F.NREC= 000030        LARGBL  003162R 025 M.BXWA= 000007         NB.DIR= 000100
FPP    = 000000          F.OPN  = 000010        LARGCV  003176R 025 M.CEI  = 177777         NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030        LARGFL  003160R 025 M.CHR  = 000202         NB.SD1= 000400
FREBLK  007400R 025 F.PREV= 000000        LARGLE  003174R 025 M.CMT  = 000206         NB.SD2= 001000
FUZZ    000142R 025 F.PRNT= 100000        LARGLN  003166R 025 M.DOUT= 000006         NB.SNM= 000040
FXPTR1  003076R 025 F.PROT= 010000        LARGPT  003164R 025 M.DQAD= 000006         NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016        LARGSF  003172R 025 M.ENCD= 000004         NB.SVR= 000010
F$EVI  = 000200          F.RATT= 000001        LARGTY  003170R 025 M.EXE  = 000000         NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034        LARGVA  003200R 025 M.EXP  = 000001         NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017        LBINC   005100R 025 M.FACT= 000001         ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000        LBUFF   003316R 025 M.FIL  = 000177         ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040        LBUFLN= 000621          M.FLR  = 177777         ND.NAM= 000004
F.BGFS= 000062          F.RES  = 000040        LCOUNT  005106R 025 M.FMT  = 000004         ND.PPN= 000100
F.BDB  = 000070          F.RET  = 177772        LENERR= 104412          M.GDUP= 000003         ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002        LENTAB  000054R 025 M.GDWN= 000003         ND.$ES= 000020
F.BITS= 177774          F.RTYP= 000000        LEXPTR  003100R 025 M.GETI= 000006         ND.$NM= 000000
F.BKDN= 000026          F.R2   = 000014        LFTOP   005066R 025 M.IBM  = 000003         ND.$NS= 000040
F.BKDS= 000020          F.R3   = 000016        LINC    005102R 025 M.IDN  = 000000         ND.$PN= 001400
F.BKEF= 000050          F.R4   = 000020        LINE    005016R 025 M.INSP= 000006         NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100        LININC  005026R 025 M.IOTA= 000000         NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072        LISIZE= 000020          M.LIT  = 000205         NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074        LITOP   005072R 025 M.LN   = 000001         NINSTK  003304R 025
F.BR   = 000020          F.STAT= 000036        LNLNG  = 104541          M.MMD  = 000010         NINTMP  003310R 025
F.CHR  = 000075          F.SVA  = 000022        LOCLCT  005036R 025 M.NEG  = 000000         NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004        LOWCOR  000002R 025 M.NOT  = 000000         NOTAVL= 104472
F.CODE= 000010          F.STP  = 000001        LSTERP  004766R 025 M.NUB  = 000006         NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002        LWIDTH  000070R 025 M.NUM  = 000204         NPRT1   004762R 025
F.DSPT= 000044          F.TRC  = 000002        L.BITS= 000003          M.OUTR= 000012         NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136        L.CNT  = 000002          M.PIRM= 000000         NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020        L.PRNT= 000001          M.PRDS= 000006         NUMDFL  003230R 025
F.EFN  = 000050          F.VAL  = 177776        L.PTR  = 000000          M.QCOD= 000000         NUMEFL  003233R 025
F.END  = 002000          F.VBN  = 000064        L.PTR1= 000000          M.QUE  = 000000         NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060        L.SUBS= 000000          M.RAND= 000012         NUMFLG  003230R 025
F.ERR  = 000052          GNOME   003102R 025 MACHIN= 000106          M.RAVL= 000000         NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025 MAXCHN= 000014          M.RCIP= 000000         NUMNEG  003232R 025
F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011         NUMQFL  003234R 025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R 025 M.REVS= 000003         NUMRIG  003231R 025
F.FLB  = 000100          HSHSIZ= 000200          MONLEN  003130R 025 M.ROLL= 000000         NUMWID  003226R 025
F.FNAM= 000110          ICHAN   005050R 025 MONPTR  003122R 025 M.SATA= 000001         NUMZFL  003235R 025
```

ERASE MACRO M1113 08-FEB-80 11:55 PAGE 46-3
SYMBOL TABLE

```
NYIERR= 104432          RARGCV  003220R   025 R.PREV= 000004          SY.CLR= 000040          T.LIT = 000020
N.BLSZ= 000036          RARGFL  003202R   025 R.PTR = 000000          SY.ERP= 000020          T.MON = 000010
N.DESC= 000020          RARGLE  003216R   025 R.SEQ = 000003          SY.NOU= 000100          T.NIL = 000004
N.DEV = 000032          RARGLN  003210R   025 R.VAR = 000002          SY.OPN= 000002          T.NUM = 002000
N.DEVN= 000034          RARGPT  003206R   025 R6    =%000006          SY.PMT= 000400          T.OPFN= 000001
N.DID = 000024          RARGSF  003214R   025 R7    =%000007          SY.SRT= 000200          T.OTHR= 040000
N.DVNM= 000032          RARGTY  003212R   025 SI    =%000005          SY.SVC= 000004          T.SOP = 000002
N.EXT = 000014          RARGVA  003222R   025 SIDAMN= 104433          SY.SVL= 000200          T.SPOP= 000040
N.FID = 000000          RBASE   005074R   025 SIFRAM  003314R  025 S.AB  = 177776          T.TMP = 000200
N.FNAM= 000006          RCOUNR  005112R   025 SISTAK  006350R  025 S.ASB = 100000          T.VAL = 100000
N.FTYP= 000014          RCOUNT  005110R   025 SITOP   006346R  025 S.BFHD= 000020          T.VAR = 001000
N.FVER= 000016          REDCNT  005070R   025 SITP    007144R  025 S.BITS= 000002          UPPER = 177400
N.NAME= 000006          REDIGN  005066R   025 SIWARN= 000200          S.CHR1= 000003          UTNAME  000344R  023
N.NEXT= 000022          REDOFF  005072R   025 SIZTAB  000044R  025 S.FATT= 000016          VALERR= 104413
N.PPN = 000024          REENTR  003060R   025 SI.BIT= 177774          S.FDB = 000140          VARBLE  007400R  025
N.STAT= 000020          RESBIT  003142R   025 SI.LEN= 177772          S.FLB = 040000          VARBLS  000000R  025
N.UNIT= 000034          RESCVT  003152R   025 SI.RET= 177772          S.FNAM= 000006          VERNUM  000000R  025
OBUFLN= 001000          RESERR= 104440          SI.VAL= 177776          S.FNB = 000036          VRSION= 000002
OCHAN   005052R  025   RESLEN  003150R   025 SKPSIZ  005134R  025 S.FNBW= 000017          WIDTH   000072R  025
OMODE   000402R  023   RESPTR  003140R   025 SPHOLD  000004R  024 S.FNTY= 000004          WROUND  000020R  025
OPTMP0  005066R  025   RESSFT  003146R   025 SPINIT  000002R  024 S.FTYP= 000002          WSDATE  000136R  025
OPTMP1  005070R  025   RESTYP  003144R   025 STKINC= 000774          S.LEN = 000002          WSFULL= 104403
OPTMP2  005072R  025   RESULT  003136R   025 STOP  = 104420          S.LINK= 177774          WSNAME  000074R  025
OPTMP3  005074R  025   RESVAL  003154R   025 SUBCNT  003106R  025 S.NFEN= 000020          WSTIME  000132R  025
OPTMP4  005076R  025   RETCOR= ****** GX         SUBDCT  003106R  025 S.TYPE= 140000          $FIS  = ****** G
OPTMP5  005100R  025   RINC    005104R   025 SUBPTR  003110R  025 S.VAL = 000000          $FPP  = ****** G
OPTMP6  005102R  025   RLTBAS  005066R   025 SYMB1   005344R  025 TKDIMS  005072R  025 $FPPLN= ****** G
OPTMP7  005104R  025   ROTINC  005070R   025 SYMF    005340R  025 TKFRAM  005066R  025 $MACHN= ****** G
OPTMP8  005106R  025   ROUND   000010R   025 SYMPT1  005346R  025 TKLEFT  005070R  025 $RSTS = ****** G
OPTMP9  005110R  025   ROVER   003312R   025 SYMTAB  005352R  025 TMODE   000403R  023 $RSX  = ****** G
OPTM10  005112R  025   RSTS  = 000000          SYMTBF= 104404          TMPTAG= 000166          $RT11 = ****** G
OPTM14  005026R  025   RSX   = 000115          SYMTOP  005350R  025 TTYBUF  004140R  025 .BOOLE= 000002
ORIGIN  000030R  025   RTOP    005070R   025 SYMTP   006272R  025 TTYLEN= 000622          .CHARA= 000000
OUTPTR  000066R  025   RT11  = 000000          SYMURN= 000054          TURKEY= 104436          .DEF. = 000002
OVROPR= 000000          R.BITS= 000003          SYNERR= 104407          TYPE  = 000166          .FLOAT= 000006
PATCH   007344R  025   R.FIX = 000001          SYSERR= 104402          T.BOL = 020000          .FSRCB= ****** G
PRECSN  000052R  025   R.LCT = 000002          SYSTAT  000000R  024 T.CON = 004000          .INTEG= 000004
RANDAC  003064R  025   R.LN  = 000002          SY.BAD= 000001          T.DIM = 000100          .MOLUN  000404R  023
RANDY   003062R  025   R.LNUM= 000010          SY.BRK= 100000          T.FUN = 010000          ...GBL= 000000
RANERR= 104411          R.NEXT= 000006          SY.BS = 000010          T.LIST= 000400          ...TPC= 000020
RARGBL  003204R  025   R.OFF = 000000

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

C 2

```
ERASE   MACRO M1113  08-FEB-80 11:55  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000174      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12928 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:40
ERASE,DM1:[305,301]ERASE/-SP=APLRX1,APLDEF,ERASE,END
```

D 3

D 2

FILEID**ERASEC

```
EEEEEEEEEE   RRRRRRRR      AAAAAA      SSSSSSSS   EEEEEEEEEE     CCCCCCCC
EEEEEEEEEE   RRRRRRRR      AAAAAA      SSSSSSSS   EEEEEEEEEE     CCCCCCCC
EE           RR      RR   AA    AA    SS         EE            CC
EE           RR      RR   AA    AA    SS         EE            CC
EE           RR      RR   AA    AA    SS         EE            CC
EE           RR      RR   AA    AA    SS         EE            CC
EEEEEEEE     RRRRRRRR     AA    AA      SSSSSS   EEEEEEEE      CC
EEEEEEEE     RRRRRRRR     AA    AA      SSSSSS   EEEEEEEE      CC
EE           RR  RR       AAAAAAAAAA          SS EE            CC
EE           RR  RR       AAAAAAAAAA          SS EE            CC
EE           RR    RR     AA    AA            SS EE            CC                ....
EE           RR    RR     AA    AA            SS EE            CC                ....
EEEEEEEEEE   RR      RR   AA    AA    SSSSSSSS   EEEEEEEEEE     CCCCCCCC         ....
EEEEEEEEEE   RR      RR   AA    AA    SSSSSSSS   EEEEEEEEEE     CCCCCCCC         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

E 2

.MAIN. MACRO M1113 08-FEB-80 11:57
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    COMMANDS                ERASEC
```

ERASEC MACRO M1113 08-FEB-80 11:57 PAGE 1

```
                        000000          RT11    =  0            ;NOT RT-11
                        000000          RSTS    =  0            ;NOT RSTS/E
                        000115          RSX     = 'M            ;RSX-11/M

                        000000          EIS     =  0            ;NO EIS
                        000000          FIS     =  0            ;NO FIS
                        000000          FPP     =  0            ;NO FPP

                        000004          FPPLEN  =  4.           ;SINGLE PRECISION

                        000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
ERASEC  MACRO M1113  08-FEB-80 11:57  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  ERASEC
                                        .IDENT  \V02.03\
                                        .SBTTL  COMMANDS                ERASEC
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       'ERASEC' is the command used to wipe out variables, functions,
                                        ;       and groups. This is simply done by calling ERASE to terminate
                                        ;       the existance of the wretched little blocks.
                                        ;
                                        ; ROUTINES CALLED:
                                        ;       BLDLST  - Builds a list of names on the SI for ERASEC to use
                                        ;       ERASE   - Does all the work to erase
                                        ;       GLBSCH  - Global search
                                        ;       NAMOUT  - Prints the name that wasn't erased
                                        ;       NEWLIN  - Prints a <CR><LF>
                                        ;       PRINTM  - Prints the 'NOT ERASED:' message
                                        ;--
000000                                  .PSECT  CODE

000000                                  ERASEC::
000000                                          CALL    BLDLST          ;Build a list of symbols on the SI
000004  016502  177776                  1$:     MOV     SI.VAL(SI),R2   ;Get a pointer to the list block
000010  116203  000002                          MOVB    L.CNT(R2),R3    ;Get the count
000014  001476                                  BEQ     7$              ;Really shouldn't happen
000016  011202                                  MOV     (R2),R2         ;Point to 1st element

000020  012204                          2$:     MOV     (R2)+,R4        ;Get pointer to symbol table entry
000022                                          CALL    GLBSCH          ;Find corresponding global
000026  001454                                  BEQ     6$              ;Is none, don't try to erase
000030  011001                                  MOV     (R0),R1         ;Point to the item
000032  032761  140000  000002                  BIT     #S.ASB!S.FLB,2.(R1)
                                                                        ;Is it a list?
000040  001023                                  BNE     4$              ;Nope
000042  005765  177774                          TST     SI.BIT(SI)      ;Did we recure once already?
000046  001415                                  BEQ     3$              ;Yes, don't do it again
000050  012725  000000G                        MOV     #SYSER,(SI)+    ;Build a SI frame
000054  005025                                  CLR     (SI)+           ;No bits (Don't really kill)
000056  010125                                  MOV     R1,(SI)+        ;Point to the list
000060                                          PUSH    <R3,R2,R0>      ;Save the registers
000066                                          CALL    1$              ;Recure (Only once!)
000072                                          POP     <R0,R2,R3>      ;Restore the Registers
000100  000403                                  BR      4$              ;    and do the erase

000102  020176  000002          3$:     CMP     R1,@2(SP)       ;Is it the same list as we are?
000106  001024                          BNE     6$              ;Yes, don't erase it
000110                          4$:     CALL    ERASE           ;Erase it, wipe it out
000114  001421                          BEQ     6$              ;Success, no problem
000116  013701  005052'                 MOV     OCHAN,R1        ;Point to the open channel
000122  026161  177772  177770          CMP     C.WIDR(R1),C.WIDC(R1)
                                                                ;Beginning of the line?
000130  001004                          BNE     5$              ;No, don't print promot msg
000132                                  QPRINT  NOTERS          ;YES, PRINT 'NOT ERASED: '
000132  012701  000000'                 MOV #NOTERS,R1          ;POINT TO NOTERS
000142  112700  000011          5$:     MOVB    #CHR.TB,R0      ;Get a tab
000146                                  CALL    OUTCHR          ;Print it
```

J 2

```
ERASEC  MACRO M1113  08-FEB-80 11:57  PAGE 44-1
COMMANDS                ERASEC

     57 000152  010401                          MOV     R4,R1           ;POINT TO NAME
     58 000154                                  CALL    NAMOUT          ;PRINT FN'S NAME
     59 000160                          6$:     SOB     R3,2$           ;Loop for each element
     60 000164  005765  177774                  TST     SI.BIT(SI)      ;Recured once?
     61 000170  001410                          BEQ     7$              ;Yes, don't print <CR><LF>
     62 000172  013701  005052'                 MOV     OCHAN,R1        ;Point to the output channel
     63 000176  026161  177772  177770          CMP     C.WIDR(R1),C.WIDC(R1)
     64                                                                 ;Beginning of line?
     65 000204  001402                          BEQ     7$              ;Yes
     66 000206                                  CALL    NEWLIN          ;No, print a <CR><LF>
     67 000212                          7$:     CALLR   KILL            ;All finished
     68
     69 000000                                  .PSECT  TABLES
     70
     71 000000      156     157     164 NOTERS: .ASCIZ  /not erased>/
        000003      040     145     162
        000006      141     163     145
        000011      144     076     000
     72                                         .EVEN
     73
```

```
ERASEC  MACRO M1113  08-FEB-80 11:57  PAGE 46
COMMANDS                ERASEC

      1          000001                              .END
```

```
ERASEC  MACRO M1113  08-FEB-80 11:57  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRFRO= 000135  C.BLOK= 000064  D.DECD= 000002  EIS    = 000000
AC1    =%000001          CHRLBR= 000073  C.BLSZ= 000140  D.DIV = 000000  ELESIZ  005132R     025
AC2    =%000002          CHRLPR= 000072  C.BUFF= 177760  D.DMD = 000010  ENDCOR  000004R     025
AC3    =%000003          CHRMNS= 000137  C.BUFL= 177762  D.DQAD= 177777  EOF    = 104475
AC4    =%000004          CHRMUL= 000075  C.CCNT= 177766  D.DROP= 000002  ERASE  = ****** GX
AC5    =%000005          CHRNEG= 000100  C.CPTR= 177764  D.ENCD= 000002  ERASEC  000000RG     020
ALLINT  003274R     025  CHRPDL= 000271  C.CZ   = 001000  D.EQ   = 000000  ERRMAX= 000166
ARYRET  003276R     025  CHRPLU= 000055  C.DIG  = 000040  D.EXPD= 000002  ERRNUM  004772R     025
ASCQUO= 000042           CHRPWR= 000120  C.FASW= 020000  D.FGET= 000005  ERRPTR  004770R     025
A.ALEN= 000004           CHRQST= 000121  C.FLEN= 000010  D.FMT  = 000004  ERRO    004774R     025
A.ARG  = 000040          CHRQUO= 000113  C.FLSH= 004000  D.FPNT= 000005  EXEBUF  005054R     025
A.ASB  = 000200          CHRRBR= 000047  C.FRC  = 000001  D.FPUT= 000005  E$CT    005056R     025
A.BITS= 000003           CHRRPR= 000042  C.IDEN= 000010  D.GE   = 000000  E$PT    005060R     025
A.CON  = 000020          CHRSEM= 000074  C.NAME= 000102  D.GT   = 000000  FA.APD= 000100
A.DIMS= 000002           CHRSMO= 000112  C.NOBF= 000400  D.IBM  = 000002  FA.CRE= 000010
A.DIM1= 000006           CHRSPA= 000040  C.NUM  = 000020  D.IND  = 000002  FA.DLK= 001000
A.NUM  = 000006          CHRUPA= 000131  C.OP   = 000004  D.INNR= 000013  FA.ENB= 100000
A.PTR  = 000000          CHR.BS= 000010  C.OPN  = 010000  D.INTR= 000006  FA.EXC= 002000
A.TMP  = 000010          CHR.CC= 000003  C.RDLN= 177776  D.LE   = 000000  FA.EXT= 000004
BADDIR= 104463           CHR.CR= 000015  C.REWT= 002000  D.LOG  = 000001  FA.NSP= 000100
BADERR= 104437           CHR.CZ= 000032  C.RND  = 000010  D.LT   = 000000  FA.NSR= 000200
BLDLST= ****** GX        CHR.GT= 000046  C.SBIT= 177756  D.MAX  = 000000  FA.POS= 010000
BLZBUB  005116R     025  CHR.LF= 000012  C.SPFX= 000100  D.MEM  = 000002  FA.RD  = 000001
BRSELF  005046R     025  CHR.LT= 000043  C.SPGL= 000200  D.MIN  = 000000  FA.RWD= 004000
CCNT    005072R     025  CHR.SI= 000017  C.SPLX= 000001  D.MUL  = 000000  FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057  C.SPRS= 000002  D.NE   = 000000  FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016  C.SRC  = 177774  D.NN   = 000000  FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011  C.SRCP= 000024  D.NR   = 000000  FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106  C.TT   = 000004  D.OR   = 000000  FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141  C.WIDC= 177770  D.PLUS= 000000  FD.BLK= 000010
CHAN04  001056R     025  CHR..E= 000145  C.WIDR= 177772  D.PSUB= 000006  FD.CCL= 000002
CHAN05  001240R     025  CHR..S= 000163  DEBUG  = 000000  D.PWR  = 000001  FD.COM= 020000
CHAN06  001422R     025  CHR..T= 000164  DEFERR= 104405  D.QQAD= 000000  FD.CR  = 000002
CHAN07  001604R     025  CHR..U= 000165  DEVIL   005114R     025 D.QUAD= 000000  FD.DIR= 000010
CHAN08  001766R     025  CHR..Z= 000172  DEVNTF= 104470  D.RES  = 000000  FD.FTN= 000001
CHAN09  002150R     025  CHR..0= 000060  DIGITS  000006R     025 D.RHO  = 000000  FD.F11= 040000
CHAN10  002332R     025  CHR..1= 000061  DIMAX  = 000020  D.ROT  = 000003  FD.INS= 000010
CHAN11  002514R     025  CH.AND= 000001  DIMBLK  003112R     025 D.SCON= 000006  FD.ISP= 002000
CHAN12  002676R     025  CMDBUF  003250R     025 DIMPTR  003114R     025 D.SDIF= 000006  FD.MNT= 100000
CHAN13  000022R     023  CMDMAX= 000010  DIMTYP  003116R     025 D.SUB  = 000000  FD.OSP= 004000
CHAN14  000204R     023  CNTSIZ  005130R     025 DIRFUL= 104466  D.SUBS= 000006  FD.PLC= 000004
CHNOPN= 104471           COMBIN  005076R     025 DOMERR= 104417  D.TAKE= 000002  FD.PRM= 000004
CHRAND= 000051           COMCNT  005126R     025 DPTERR= 104416  D.TRAN= 000004  FD.PSE= 010000
CHRBEL= 000007           COMERR= 104425  DYDCNT  003224R     025 D.UNON= 000006  FD.RAH= 000001
CHRBKR= 000133           COMPTR  005122R     025 D.AND  = 000000  D.$MAX= 000000  FD.RAN= 000002
CHRBOL= 000200           COMRTN  003156R     025 D.BXCT= 000007  D.$MIN= 000000  FD.REC= 000001
CHRBOX= 000114           COMSKP  005124R     025 D.BXIO= 000007  D.$MUL= 000000  FD.RWM= 000001
CHRCMA= 000054           CORPTR  005120R     025 D.BXNL= 000007  D.$PLU= 000000  FD.SDI= 000020
CHRCOL= 000076           CS.ASS= 000200      025 D.BXPP= 000007  D.$RES= 000000  FD.SQD= 000040
CHRDEL= 000107           CS.BOO= 000010          D.BXPW= 000007  D.$SUB= 000000  FD.TTY= 000004
CHRDIV= 000053           CS.FLO= 000020          D.BXRL= 000007  D.1COM= 000001  FD.WBH= 000002
CHRDLR= 000176           CS.IDN= 000040          D.CIRC= 000001  D.1EXP= 000002  FF.CHR= 000005
CHRDLT= 000110           CS.ONE= 000100          D.COMB= 000001  D.1ROT= 000003  FF.NV  = 000003
CHRDOT= 000056           CTEPS   000152R     025 D.COMP= 000001  EDICHR  005063R     025 FF.POE= 000002
CHREP  = 000105          CTMAX   000212R     025 D.CONT= 000006  EDICNT  005064R     025 FF.RWD= 000001
CHRERR= 104431           CTRMAX  000172R     025 D.CTLM= 000000  EDIFLG  005062R     025 FF.RWF= 000006
CHRFIL= 000177           C.ALOC= 000162          D.DEAL= 000003  EDIT   = 000003          FF.SPC= 000004
```

SYMBOL TABLE

```
FHEADR  005012R     025 F.FLB = 000100      HSHMSK= 177601          MONBLK  003120R     025 M.RDCT= 000011
FHRET   005040R     025 F.FNAM= 000110      HSHSIZ= 000200          MONCVT  003132R     025 M.REVS= 000003
FILNTF= 104467          F.FNB = 000102      ICHAN   005050R     025 MONLEN  003130R     025 M.ROLL= 000000
FIS   = 000000          F.FNDF= 000010      ILFNAM= 104464          MONPTR  003122R     025 M.SATA= 000001
FISCON= 046000          F.FTYP= 000116      ILLCHN= 104540          MONSFT  003126R     025 M.SCAN= 000011
FIXPTR  003074R     025 F.FUN = 000010      ILLSW = 104565          MONTYP  003124R     025 M.SHAP= 000000
FLAGS   004764R     025 F.FVER= 000120      INDERR= 104410          MONVAL  003134R     025 M.SIGN= 000000
FLCBUF  003250R     025 F.HEAD= 000006      INNTMP  003250R     025 M.ABS = 000000          M.SUBR= 000001
FLINE   005014R     025 F.HIBK= 000004      INNTM2  005016R     025 M.BOL = 000200          M.SYM = 000203
FLOEXP  003270R     025 F.LBL = 000020      INPTR   000064R     025 M.BXAV= 000007          M.SYN = 000201
FLONEG  003272R     025 F.LINE= 000012      INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FLSWTB  000222R     025 F.LOC = 000004      IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FLTLEN= 000010          F.LSTC= 000004      I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FNINSI  005044R     025 F.LUN = 000042      I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FORIGN  000032R     025 F.MBCT= 000054      I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.APD= 000106          F.MBC1= 000055      I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.MFY= 000002          F.MBFG= 000056      I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.RD = 000001          F.MON = 000002      I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.UPD= 000006          F.NIL = 000001      I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FO.WRT= 000016          F.NRBD= 000024      KILL  = ****** GX       M.BXPW= 000007          M.1SCN= 000011
FPMODE  000042R     025 F.NREC= 000030      LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FPP   = 000000          F.OPN = 000010      LABLPT  005042R     025 M.BXTS= 000007          NAMOUT= ****** GX
FPPLEN= 000004          F.OVBS= 000030      LARGBL  003162R     025 M.BXWA= 000007          NB.DEV= 000200
FREBLK  007400R     025 F.PREV= 000000      LARGCV  003176R     025 M.CEI = 177777          NB.DIR= 000100
FUZZ    000142R     025 F.PRNT= 100000      LARGFL  003160R     025 M.CHR = 000202          NB.NAM= 000004
FXPTR1  003076R     025 F.PROT= 010000      LARGLE  003174R     025 M.CMT = 000206          NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016      LARGLN  003166R     025 M.DOUT= 000006          NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001      LARGPT  003164R     025 M.DQAD= 000006          NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034      LARGSF  003172R     025 M.ENCD= 000004          NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017      LARGTY  003170R     025 M.EXE = 000000          NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000      LARGVA  003200R     025 M.EXP = 000001          NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040      LBINC   005100R     025 M.FACT= 000001          NB.VER= 000001
F.BBFS= 000062          F.RES = 000040      LBUFF   003316R     025 M.FIL = 000177          ND.DEV= 000200
F.BDB = 000070          F.RET = 177772      LBUFLN= 000621          M.FLR = 177777          ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002      LCOUNT  005106R     025 M.FMT = 000004          ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000      LENERR= 104412          M.GDUP= 000003          ND.PPN= 000100
F.BKDN= 000026          F.R2  = 000014      LENTAB  000054R     025 M.GDWN= 000003          ND.$EX= 000000
F.BKDS= 000020          F.R3  = 000016      LEXPTR  003100R     025 M.GETI= 000006          ND.$ES= 000020
F.BKEF= 000050          F.R4  = 000020      LFTOP   005066R     025 M.IBM = 000003          ND.$NM= 000000
F.BKP1= 000051          F.SEQN= 000100      LINC    005102R     025 M.IDN = 000000          ND.$NS= 000040
F.BKST= 000024          F.SPDV= 000072      LINE    005016R     025 M.INSP= 000006          ND.$PN= 001400
F.BKVB= 000064          F.SPUN= 000074      LININC  005026R     025 M.IOTA= 000000          NEGSGN  005063R     025
F.BR  = 000020          F.STBK= 000036      LISIZE= 000020          M.LIT = 000205          NEWLIN= ****** GX
F.CHR = 000075          F.STK = 000022      LITOP   005072R     025 M.LN  = 000001          NINPT1  003302R     025
F.CNTG= 000034          F.STOP= 000004      LNLNG = 104541          M.MMD = 000010          NINPT2  003300R     025
F.CODE= 000010          F.STP = 000001      LOCLCT  005036R     025 M.NEG = 000000          NINSTK  003304R     025
F.DFNB= 000046          F.SUSP= 000002      LOWCOR  000002R     025 M.NOT = 000000          NINTMP  003310R     025
F.DSPT= 000044          F.TRC = 000002      LSTERP  004766R     025 M.NUB = 000006          NINTYP  003306R     025
F.DVNM= 000134          F.UNIT= 000136      LWIDTH  000070R     025 M.NUM = 000204          NOTAVL= 104472
F.EFBK= 000010          F.URBD= 000020      L.BITS= 000003          M.OUTR= 000012          NOTERS  000000R     002
F.EFN = 000050          F.VAL = 177776      L.CNT = 000002          M.PITM= 000000          NOTOPN= 104473
F.END = 002000          F.VBN = 000064      L.PRNT= 000001          M.PRDS= 000006          NPRT1   004762R     025
F.EOBB= 000032          F.VBSZ= 000060      L.PTR = 000000          M.QQAD= 000000          NULL    007364R     025
F.ERR = 000052          GLBSCH= ****** GX   L.PTR1= 000004          M.QUAD= 000000          NUMCHN= 000020
F.FACC= 000043          GNOME   003102R     025 L.SUBS= 000002      M.RAND= 000012          NUMDFL  003230R     025
F.FFBY= 000014          HASHC   005136R     025 MACHIN= 000106      M.RAVL= 000000          NUMEFL  003233R     025
F.FLAG= 000002          HASHTB  005140R     025 MAXCHN= 000014      M.RCIP= 000000          NUMESN  003236R     025
```

SYMBOL TABLE

```
NUMFLG  003230R     025 PATCH   007344R     025 R.LCT = 000002          SYSTAT  000000R     024 T.FUN = 01000C
NUMLEN  000062R     025 PRECSN  000052R     025 R.LN  = 000002          SY.BAD= 000001          T.LIST= 000400
NUMNEG  003232R     025 PRINTM= ****** GX       R.LNUM= 000010          SY.BRK= 100000          T.LIT = 000020
NUMQFL  003234R     025 RANDAC  003064R     025 R.NEXT= 000006          SY.BS = 000010          T.MON = 000010
NUMRIG  003231R     025 RANDY   003062R     025 R.OFF = 000000          SY.CLR= 000040          T.NIL = 000004
NUMWID  003226R     025 RANERR= 104411          R.PREV= 000004          SY.ERP= 000020          T.NUM = 002000
NUMZFL  003235R     025 RARGBL  003204R     025 R.PTR = 000000          SY.NOU= 000100          T.OPFN= 000001
NYIERR= 104432          RARGCV  003220R     025 R.SEQ = 000003          SY.OPN= 000002          T.OTHR= 040000
N.BLSZ= 000036          RARGFL  003202R     025 R.VAR = 000002          SY.PMT= 001000          T.SOP = 000002
N.DESC= 000020          RARGLE  003216R     025 R6    =%000006          SY.SRT= 000400          T.SPOP= 000040
N.DEV = 000032          RARGLN  003210R     025 R7    =%000007          SY.SVC= 000004          T.TMP = 000200
N.DEVN= 000034          RARGPT  003206R     025 SI    =%000005          SY.SVL= 000200          T.VAL = 100000
N.DID = 000024          RARGSF  003214R     025 SIDMN= 104433           S.AB  = 177776          T.VAR = 001000
N.DVNM= 000032          RARGTY  003212R     025 SIFRAM  003314R     025 S.ASB = 100000          UPPER = 177400
N.EXT = 000014          RARGVA  003222R     025 SISTAK  006350R     025 S.BFHD= 000020          UTNAME  000344R     023
N.FID = 000000          RBASE   005074R     025 SITOP   006346R     025 S.BITS= 000002          VALERR= 104413
N.FNAM= 000006          RCOUNR  005112R     025 SITP    007144R     025 S.CHR1= 000003          VARBLE  007400R     025
N.FTYP= 000014          RCOUNT  005110R     025 SIWARN= 000200          S.FATT= 000016          VARBLS  000000R     025
N.FVER= 000016          REDCNT  005070R     025 SIZTAB  000044R     025 S.FDB = 000140          VERNUM  000000R     025
N.NAME= 000006          REDIGN  005066R     025 SI.BIT= 177774          S.FLB = 040000          VRSION= 000002
N.NEXT= 000022          REDOFF  005072R     025 SI.LEN= 177772          S.FNAM= 000006          WIDTH   000072R     025
N.PPN = 000024          REENTR  003060R     025 SI.RET= 177772          S.FNB = 000036          WROUND  000020R     025
N.STAT= 000020          RESBIT  003142R     025 SI.VAL= 177776          S.FNBW= 000017          WSDATE  000136R     025
N.UNIT= 000034          RESCVT  003152R     025 SKPSIZ  005134R     025 S.FNTY= 000004          WSFULL= 104403
OBUFLN= 001000          RESERR= 104440          SPHOLD  000004R     024 S.FTYP= 000002          WSNAME  000074R     025
OCHAN   005052R     025 RESLEN  003150R     025 SPINIT  000002R     024 S.LEN = 000002          WSTIME  000132R     025
OMODE   0004C2R     023 RESPTR  003140R     025 STKINC= 000774          S.LINK= 177774          $FIS  = ****** G
OPTMP0  005066R     025 RESSFT  003146R     025 STOP  = 104420          S.NFEN= 000020          $FPP  = ****** G
OPTMP1  005070R     025 RESTYP  003144R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FPPLN= ****** G
OPTMP2  005072R     025 RESULT  003136R     025 SUBDCT  003106R     025 S.VAL = 000000          $MACHN= ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $RSTS = ****** G
OPTMP4  005076R     025 RINC    005104R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $RSX  = ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RT11 = ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMPT1  005346R     025 TMODE   000403R     023 .BOOLE= 000002
OPTMP7  005104R     025 ROUND   000010R     025 SYMTAB  005352R     025 TMPTAG= 000166          .CHARA= 000000
OPTMP8  005106R     025 ROVER   003312R     025 SYMTBF= 104404          TTYBUF  004140R     025 .DEF. = 000002
OPTMP9  005110R     025 RSTS  = 000000          SYMTOP  005350R     025 TTYLEN= 000622          .FLOAT= 000006
OPTM10  005112R     025 RSX   = 000115          SYMTP   006272R     025 TURKEY= 104436          .FSRCB= ****** G
OPTM14  005026R     025 RTOP    005070R     025 SYMWRN= 000054          TYPE  = 000166          .INTEG= 000004
ORIGIN  000030R     025 RT11  = 000000          SYNERR= 104407          T.BOL = 020000          .MOLUN  000404R     023
OUTCHR= ****** GX       R.BITS= 000003          SYSER = ****** GX       T.CON = 004000          ...GBL= 000000
OUTPTR  000066R     025 R.FIX = 000001          SYSERR= 104402          T.DIM = 000100          ...TPC= 000020
OVROPR= 000000

. ABS.  000000      000
        000000      001
TABLES  000014      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MUNBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
```

```
ERASEC  MACRO M1113  08-FEB-80 11:57  PAGE 46-4
SYMBOL TABLE

TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000216      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12974 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:52
ERASEC,DM1:[305,301]ERASEC/-SP=APLRX1,APLDEF,ERASEC,END
```

FILEID**ERRMSG

```
EEEEEEEEEE   RRRRRRRR     RRRRRRRR     MM      MM     SSSSSSSS     GGGGGGGG
EEEEEEEEEE   RRRRRRRR     RRRRRRRR     MM      MM     SSSSSSSS     GGGGGGGG
EE           RR      RR   RR      RR   MMMM  MMMM   SS           GG
EE           RR      RR   RR      RR   MMMM  MMMM   SS           GG
EE           RR      RR   RR      RR   MM  MM  MM   SS           GG
EE           RR      RR   RR      RR   MM  MM  MM   SS           GG
EEEEEEEE     RRRRRRRR     RRRRRRRR     MM      MM     SSSSSS     GG
EEEEEEEE     RRRRRRRR     RRRRRRRR     MM      MM     SSSSSS     GG
EE           RR  RR       RR  RR       MM      MM           SS   GG  GGGGGG
EE           RR  RR       RR  RR       MM      MM           SS   GG  GGGGGG
EE           RR    RR     RR    RR     MM      MM           SS   GG      GG      ....
EE           RR    RR     RR    RR     MM      MM           SS   GG      GG      ....
EEEEEEEEEE   RR      RR   RR      RR   MM      MM   SSSSSSSS     GGGGGG        ....
EEEEEEEEEE   RR      RR   RR      RR   MM      MM   SSSSSSSS     GGGGGG        ....

LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .TITLE  QIOMAC - QIOSYM MACRO DEFINITION
                                ;
                                ; DATE OF LAST MODIFICATION:
                                ;
                                ;       C.A. D'ELIA     19-NOV-77
                                ;
                                ;
                                ; ***** ALWAYS UPDATE THE FOLLOWING TWO LINES TOGETHER
                                ;       .IDENT  /0331/
000331                          ;       QI.VER=0331
                                ;
                                ;
                                ; COPYRIGHT (C) 1973,1977
                                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                                ;
                                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
                                ; SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
                                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR
                                ; ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE
                                ; MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH
                                ; SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE
                                ; TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
                                ; IN DEC.
                                ;
                                ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
                                ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                                ; EQUIPMENT CORPORATION.
                                ;
                                ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
                                ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
                                ;
                                ;
                                ; PETER H. LIPMAN 1-OCT-73
                                ;
                                ;+
                                ; MACRO TO DEFINE STANDARD QUEUE I/O DIRECTIVE FUNCTION VALUES
                                ; AND IOSB RETURN VALUES.  TO INVOKE AT ASSEMBLY TIME (WITH LOCAL
                                ; DEFINITION) USE:
                                ;
                                ;       QIOSY$          ;DEFINE SYMBOLS
                                ;
                                ; TO OBTAIN GLOBAL DEFINITION OF THESE SYMBOLS USE:
                                ;
                                ;       QIOSY$ DEF$G    ;SYMBOLS DEFINED GLOBALLY
                                ;
                                ; THE MACRO CAN BE CALLED ONCE ONLY AND THEN
                                ; REDEFINES ITSELF AS NULL.
                                ;-
                                ;
                                        .MACRO  QIOSY$  $$$GBL,$$$MSG
                                        .IIF    IDN,<$$$GBL>,<DEF$G>,   .GLOBL  QI.VER
                                        .IF     IDN,<$$$MSG>,<DEF$S>
                                        $$$MAX=0
                                        $$MSG=1
                                        .IFF
                                        $$MSG=0
                                        .ENDC
```

```
                                .MCALL  IOERR$
                                IOERR$  $$$GBL          ;I/O ERROR CODES FROM HANDLERS, FCP, FCS
                                .MCALL  DRERR$
                                DRERR$  $$$GBL          ;DIRECTIVE STATUS WORD ERROR CODES
                                .IF     DIF,<$$$MSG>,<DEF$S>
                                .MCALL  FILIO$
                                FILIO$  $$$GBL          ;DEFINE GENERAL I/O FUNCTION CODES
                                .MCALL  SPCIO$
                                SPCIO$  $$$GBL          ;DEVICE DEPENDENT I/O FUNCTION CODES
                                .MACRO  QIOSY$  ARG,ARG1,ARG2   ;RECLAIM MACRO STORAGE
                                .ENDM   QIOSY$
                                .ENDC
                                .ENDM   QIOSY$
```

```
                        ;
                        ; DEFINE THE ERROR CODES RETURNED BY DEVICE HANDLER AND FILE PRIMITIVES
                        ; IN THE FIRST WORD OF THE I/O STATUS BLOCK
                        ; THESE CODES ARE ALSO RETURNED BY FILE CONTROL SERVICES (FCS) IN THE
                        ; BYTE F.ERR IN THE FILE DESCRIPTOR BLOCK (FDB)
                        ;       THE BYTE F.ERR+1 IS 0 IF F.ERR CONTAINS A HANDLER OR FCP ERROR CODE.
                        ;
                                .MACRO  IOERR$  $$$GBL
                                .MCALL  .IOER.,DEFIN$
                                .IF     IDN,<$$$GBL>,<DEF$G>
                                ...GBL=1
                                .IFF
                                ...GBL=0
                                .ENDC
                                .IIF    NDF,$$MSG,$$MSG=0
                        ;
                        ; SYSTEM STANDARD CODES, USED BY EXECUTIVE AND DRIVERS
                        ;
                                .IOER.  IE.BAD,-01.,<BAD PARAMETERS>
                                .IOER.  IE.IFC,-02.,<INVALID FUNCTION CODE>
                                .IOER.  IE.DNR,-03.,<DEVICE NOT READY>
                                .IOER.  IE.VER,-04.,<PARITY ERROR ON DEVICE>
                                .IOER.  IE.ONP,-05.,<HARDWARE OPTION NOT PRESENT>
                                .IOER.  IE.SPC,-06.,<ILLEGAL USER BUFFER>
                                .IOER.  IE.DNA,-07.,<DEVICE NOT ATTACHED>
                                .IOER.  IE.DAA,-08.,<DEVICE ALREADY ATTACHED>
                                .IOER.  IE.DUN,-09.,<DEVICE NOT ATTACHABLE>
                                .IOER.  IE.EOF,-10.,<END OF FILE DETECTED>
                                .IOER.  IE.EOV,-11.,<END OF VOLUME DETECTED>
                                .IOER.  IE.WLK,-12.,<WRITE ATTEMPTED TO LOCKED UNIT>
                                .IOER.  IE.DAO,-13.,<DATA OVERRUN>
                                .IOER.  IE.SRE,-14.,<SEND/RECEIVE FAILURE>
                                .IOER.  IE.ABO,-15.,<REQUEST TERMINATED>
                                .IOER.  IE.PRI,-16.,<PRIVILEGE VIOLATION>
                                .IOER.  IE.RSU,-17.,<SHARABLE RESOURCE IN USE>
                                .IOER.  IE.OVR,-18.,<ILLEGAL OVERLAY REQUEST>
                                .IOER.  IE.BYT,-19.,<ODD BYTE COUNT (OR VIRTUAL ADDRESS)>
                                .IOER.  IE.BLK,-20.,<LOGICAL BLOCK NUMBER TOO LARGE>
                                .IOER.  IE.MOD,-21.,<INVALID UDC MODULE #>
                                .IOER.  IE.CON,-22.,<UDC CONNECT ERROR>
                                .IOER.  IE.BBE,-56.,<BAD BLOCK ON DEVICE>
                                .IOER.  IE.STK,-58.,<NOT ENOUGH STACK SPACE (FCS OR FCP)>
                                .IOER.  IE.FHE,-59.,<FATAL HARDWARE ERROR ON DEVICE>
                                .IOER.  IE.EOT,-62.,<END OF TAPE DETECTED>
                                .IOER.  IE.OFL,-65.,<DEVICE OFF LINE>
                                .IOER.  IE.BCC,-66.,<BLOCK CHECK, CRC, OR FRAMING ERROR>

                        ;
                        ; FILE PRIMITIVE CODES
                        ;

                                .IOER.  IE.NOD,-23.,<CALLER'S NODES EXHAUSTED>
                                .IOER.  IE.DFU,-24.,<DEVICE FULL>
                                .IOER.  IE.IFU,-25.,<INDEX FILE FULL>
                                .IOER.  IE.NSF,-26.,<NO SUCH FILE>
```

```
                                        .IOER.  IE.LCK,-27.,<LOCKED FROM READ/WRITE ACCESS>
                                        .IOER.  IE.HFU,-28.,<FILE HEADER FULL>
                                        .IOER.  IE.WAC,-29.,<ACCESSED FOR WRITE>
                                        .IOER.  IE.CKS,-30.,<FILE HEADER CHECKSUM FAILURE>
                                        .IOER.  IE.WAT,-31.,<ATTRIBUTE CONTROL LIST FORMAT ERROR>
                                        .IOER.  IE.RER,-32.,<FILE PROCESSOR DEVICE READ ERROR>
                                        .IOER.  IE.WER,-33.,<FILE PROCESSOR DEVICE WRITE ERROR>
                                        .IOER.  IE.ALN,-34.,<FILE ALREADY ACCESSED ON LUN>
                                        .IOER.  IE.SNC,-35.,<FILE ID, FILE NUMBER CHECK>
                                        .IOER.  IE.SQC,-36.,<FILE ID, SEQUENCE NUMBER CHECK>
                                        .IOER.  IE.NLN,-37.,<NO FILE ACCESSED ON LUN>
                                        .IOER.  IE.CLO,-38.,<FILE WAS NOT PROPERLY CLOSED>
                                        .IOER.  IE.DUP,-57.,<ENTER - DUPLICATE ENTRY IN DIRECTORY>
                                        .IOER.  IE.BVR,-63.,<BAD VERSION NUMBER>
                                        .IOER.  IE.BHD,-64.,<BAD FILE HEADER>
                                        .IOER.  IE.EXP,-75.,<FILE EXPIRATION DATE NOT REACHED>
                                        .IOER.  IE.BTF,-76.,<BAD TAPE FORMAT>
                                        .IOER.  IE.ALC,-84.,<ALLOCATION FAILURE>
                                        .IOER.  IE.ULK,-85.,<UNLOCK ERROR>
                                        .IOER.  IE.WCK,-86.,<WRITE CHECK FAILURE>

                                ;
                                ; FILE CONTROL SERVICES CODES
                                ;

                                        .IOER.  IE.NBF,-39.,<OPEN - NO BUFFER SPACE AVAILABLE FOR FILE>
                                        .IOER.  IE.RBG,-40.,<ILLEGAL RECORD SIZE>
                                        .IOER.  IE.NBK,-41.,<FILE EXCEEDS SPACE ALLOCATED, NO BLOCKS>
                                        .IOER.  IE.ILL,-42.,<ILLEGAL OPERATION ON FILE DESCRIPTOR BLOCK>
                                        .IOER.  IE.BTP,-43.,<BAD RECORD TYPE>
                                        .IOER.  IE.RAC,-44.,<ILLEGAL RECORD ACCESS BITS SET>
                                        .IOER.  IE.RAT,-45.,<ILLEGAL RECORD ATTRIBUTES BITS SET>
                                        .IOER.  IE.RCN,-46.,<ILLEGAL RECORD NUMBER - TOO LARGE>
                                        .IOER.  IE.2DV,-48.,<RENAME - 2 DIFFERENT DEVICES>
                                        .IOER.  IE.FEX,-49.,<RENAME - NEW FILE NAME ALREADY IN USE>
                                        .IOER.  IE.BDR,-50.,<BAD DIRECTORY FILE>
                                        .IOER.  IE.RNM,-51.,<CAN'T RENAME OLD FILE SYSTEM>
                                        .IOER.  IE.BDI,-52.,<BAD DIRECTORY SYNTAX>
                                        .IOER.  IE.FOP,-53.,<FILE ALREADY OPEN>
                                        .IOER.  IE.BNM,-54.,<BAD FILE NAME>
                                        .IOER.  IE.BDV,-55.,<BAD DEVICE NAME>
                                        .IOER.  IE.NFI,-60.,<FILE ID WAS NOT SPECIFIED>
                                        .IOER.  IE.ISQ,-61.,<ILLEGAL SEQUENTIAL OPERATION>
                                        .IOER.  IE.NNC,-77.,<NOT ANSI 'D' FORMAT BYTE COUNT>

                                ;
                                ; NETWORK ACP CODES
                                ;

                                        .IOER.  IE.AST,-80.,<NO AST SPECIFIED IN CONNECT>
                                        .IOER.  IE.NNN,-68.,<NO SUCH NODE>
                                        .IOER.  IE.NFW,-69.,<PATH LOST TO PARTNER> ;THIS CODE MUST BE ODD
                                        .IOER.  IE.BLB,-70.,<BAD LOGICAL BUFFER>
                                        .IOER.  IE.TMM,-71.,<TOO MANY OUTSTANDING MESSAGES>
                                        .IOER.  IE.NDR,-72.,<NO DYNAMIC SPACE AVAILABLE>
                                        .IOER.  IE.CNR,-73.,<CONNECTION REJECTED>
                                        .IOER.  IE.TMO,-74.,<TIMEOUT ON REQUEST>
                                        .IOER.  IE.NNL,-78.,<NOT A NETWORK LUN>
```

```
186                                     ;
187                                     ; ICS/ICR ERROR CODES
188                                     ;
189                                             .IOER.  IE.NLK,-79.,<TASK NOT LINKED TO SPECIFIED ICS/ICR INTERRUPTS>
190                                             .IOER.  IE.NST,-80.,<SPECIFIED TASK NOT INSTALLED>
191                                             .IOER.  IE.FLN,-81.,<DEVICE OFFLINE WHEN OFFLINE REQUEST WAS ISSUED>
192
193
194                                     ;
195                                     ; TTY ERROR CODES
196                                     ;
197
198                                             .IOER.  IE.IES,-82.,<INVALID ESCAPE SEQUENCE>
199                                             .IOER.  IE.PES,-83.,<PARTIAL ESCAPE SEQUENCE>
200
201
202                                     ;
203                                     ; SUCCESSFUL RETURN CODES---
204                                     ;
205
206                                             DEFIN$  IS.PND,+00.     ;OPERATION PENDING
207                                             DEFIN$  IS.SUC,+01.     ;OPERATION COMPLETE, SUCCESS
208                                             DEFIN$  IS.RDD,+02.     ;(RX11) FLOPPY DISK SUCCESSFUL COMPLETION
209                                                                     ;OF A READ PHYSICAL, AND DELETED
210                                                                     ;DATA MARK WAS SEEN IN SECTOR HEADER
211                                                                     ; LAST SECTOR.
212                                             DEFIN$  IS.BV,+05.      ;(A/D READ) AT LEAST ONE BAD VALUE
213                                                                     ;WAS READ (REMAINDER MAY BE GOOD).
214                                                                     ;BAD CHANNEL IS INDICATED BY A
215                                                                     ;NEGATIVE VALUE IN THE BUFFER.
216
217
218                                     ;
219                                     ; TTY SUCCESS CODES
220                                     ;
221
222                                             DEFIN$  IS.CR,<15*400+1>  ;CARRIAGE RETURN WAS TERMINATOR
223                                             DEFIN$  IS.ESC,<33*400+1> ;ESCAPE (ALTMODE) WAS TERMINATOR
224                                             DEFIN$  IS.CC,<3*400+1> ;CONTROL-C WAS TERMINATOR
225                                             DEFIN$  IS.ESQ,<233*400+1>  ;ESCAPE SEQUENCE WAS TERMINATOR
226                                             DEFIN$  IS.PES,<200*400+1>  ;PARTIAL ESCAPE SEQUENCE TERMINATOR
227                                             DEFIN$  IS.EOT,<4*400+1>  ;EOT WAS TERMINATOR (BLOCK MODE INPUT)
228                                             DEFIN$  IS.TAB,<11*400+1> ;TAB WAS TERMINATOR (FORMS MODE INPUT)
229                                             DEFIN$  IS.TMO,+2.      ;REQUEST TIMED OUT
230
231
232                                     ; ******
233                                     ;
234                                     ; THE NEXT AVAILABLE ERROR NUMBER IS: -87.
235                                     ; ALL (BUT TWO) LOWER NUMBERS ARE IN USE!!
236                                     ; NOTE: ERROR CODES -47. AND -67. HAVE BEEN RETIRED AND CAN
237                                     ;       BE ASSIGNED AT A LATER DATE.
238                                     ;
239                                     ; *****
240                                             .IF     EQ,$$MSG
241                                             .MACRO  IOERR$  A
242                                             .ENDM   IOERR$
```

```
                                        .ENDC
                                        .ENDM   IOERR$
```

```
                        ;
                        ; DEFINE THE DIRECTIVE ERROR CODES RETURNED IN THE DIRECTIVE STATUS WORD
                        ;
                        ;       FILE CONTROL SERVICES (FCS) RETURNS THESE CODES IN THE BYTE F.ERR
                        ; OF THE FILE DESCRIPTOR BLOCK (FDB).  TO DISTINGUISH THEM FROM THE
                        ; OVERLAPPING CODES FROM HANDLER AND FILE PRIMITIVES, THE BYTE
                        ; F.ERR+1 IN THE FDB WILL BE NEGATIVE FOR A DIRECTIVE ERROR CODE.
                        ;
                                .MACRO  DRERR$  $$$GBL
                                .MCALL  .QIOE.,DEFIN$
                                .IF     IDN,<$$$GBL>,<DEF$G>
                                ...GBL=1
                                .IFF
                                ...GBL=0
                                .ENDC
                                .IIF    NDF,$$MSG,$$MSG=0
                        ;
                        ; STANDARD ERROR CODES RETURNED BY DIRECTIVES IN THE DIRECTIVE STATUS WORD
                        ;
                                .QIOE.  IE.UPN,-01.,<INSUFFICIENT DYNAMIC STORAGE>
                                .QIOE.  IE.INS,-02.,<SPECIFIED TASK NOT INSTAL'ED>
                                .QIOE.  IE.PTS,-03.,<PARTITION TOO SMALL FOR TASK>
                                .QIOE.  IE.UNS,-04.,<INSUFFICIENT DYNAMIC STORAGE FOR SEND>
                                .QIOE.  IE.ULN,-05.,<UN-ASSIGNED LUN>
                                .QIOE.  IE.HWR,-06.,<DEVICE HANDLER NOT RESIDENT>
                                .QIOE.  IE.ACT,-07.,<TASK NOT ACTIVE>
                                .QIOE.  IE.ITS,-08.,<DIRECTIVE INCONSISTENT WITH TASK STATE>
                                .QIOE.  IE.FIX,-09.,<TASK ALREADY FIXED/UNFIXED>
                                .QIOE.  IE.CKP,-10.,<ISSUING TASK NOT CHECKPOINTABLE>
                                .QIOE.  IE.TCH,-11.,<TASK IS CHECKPOINTABLE>
                                .QIOE.  IE.RBS,-15.,<RECEIVE BUFFER IS TOO SMALL>
                                .QIOE.  IE.PRI,-16.,<PRIVILEGE VIOLATION>
                                .QIOE.  IE.RSU,-17.,<RESOURCE IN USE>
                                .QIOE.  IE.NSW,-18.,<NO SWAP SPACE AVAILABLE>
                                .QIOE.  IE.ILV,-19.,<ILLEGAL VECTOR SPECIFIED>
                        .
                        .
                        .
                                .QIOE.  IE.AST,-80.,<DIRECTIVE ISSU D/NOT ISSUED FROM AST>
                                .QIOE.  IE.MAP,-81.,<ILLEGAL MAPPING SPECIFIED>
                                .QIOE.  IE.IOP,-83.,<WINDOW HAS I/O IN PROGRESS>
                                .QIOE.  IE.ALG,-84.,<ALIGNMENT ERROR>
                                .QIOE.  IE.WOV,-85.,<ADDRESS WINDOW ALLOCATION OVERFLOW>
                                .QIOE.  IE.NVR,-86.,<INVALID REGION ID>
                                .QIOE.  IE.NVW,-87.,<INVALID ADDRESS WINDOW ID>
                                .QIOE.  IE.ITP,-88.,<INVALID TI PARAMETER>
                                .QIOE.  IE.IBS,-89.,<INVALID SEND BUFFER SIZE ( .GT. 255.)>
                                .QIOE.  IE.LNL,-90.,<LUN LOCKED IN USE>
                                .QIOE.  IE.IUI,-91.,<INVALID UIC>
                                .QIOE.  IE.IDU,-92.,<INVALID DEVICE OR UNIT>
                                .QIOE.  IE.ITI,-93.,<INVALID TIME PARAM  RS>
                                .QIOE.  IE.PNS,-94.,<PARTITION/REGION NO. IN SYSTEM>
                                .QIOE.  IE.IPR,-95.,<INVALID PRIORITY ( .GT. 250.)>
                                .QIOE.  IE.ILU,-96.,<INVALID LUN>
                                .QIOE.  IE.IEF,-97.,<INVALID EVENT FLAG ( .GT. 64.)>
                                .QIOE.  IE.ADP,-98.,<PART OF DPB OUT OF USER'S SPACE>
                                .QIOE.  IE.SDP,-99.,<DIC OR DPB SIZE INVALID>
```

```
303                             ;
304                             ; SUCCESS CODES FROM DIRECTIVES - PLACED IN THE DIRECTIVE STATUS WORD
305                             ;
306                                   DEFIN$  IS.CLR,0        ;EVENT FLAG WAS CLEAR
307                                                           ;FROM CLEAR EVENT FLAG DIRECTIVE
308                                   DEFIN$  IS.SET,2        ;EVENT FLAG WAS SET
309                                                           ;FROM SET EVENT FLAG DIRECTIVE
310                                   DEFIN$  IS.SPD,2        ;TASK WAS SUSPENDED
311                             ;
312                             ;
313                                   .IF     EQ,$$MSG
314                                   .MACRO  DRERR$  A
315                                   .ENDM   DRERR$
316                                   .ENDC
317                                   .ENDM   DRERR$
```

```
                        ;
                        ; DEFINE THE GENERAL I/O FUNCTION CODES - DEVICE INDEPENDENT
                        ;
                                .MACRO  FILIO$  $$$GBL
                                .MCALL  .WORD.,DEFIN$
                                .IF     IDN,<$$$GBL>,<DEF$G>
                                ...GBL=1
                                .IFF
                                ...GBL=0
                                .ENDC
                        ;
                        ; GENERAL I/O QUALIFIER BYTE DEFINITIONS
                        ;
                                .WORD.  IQ.X,001,000    ;NO ERROR RECOVERY
                                .WORD.  IQ.Q,002,000    ;QUEUE REQUEST IN EXPRESS QUEUE
                                .WORD.  IQ.S,004,000    ;SYNONYM FOR IQ.UMD
                                .WORD.  IQ.UMD,004,000  ;USER MODE DIAGNOSTIC STATUS REQUIRED
                        ;
                        ; EXPRESS QUEUE COMMANDS
                        ;
                                .WORD.  IO.KIL,012,000  ;KILL CURRENT REQUEST
                                .WORD.  IO.RDN,022,000  ;I/O RUNDOWN
                                .WORD.  IO.UNL,042,000  ;UNLOAD I/O HANDLER TASK
                                .WORD.  IO.LTK,050,000  ;LOAD A TASK IMAGE FILE
                                .WORD.  IO.RTK,060,000  ;RECORD A TASK IMAGE FILE
                                .WORD.  IO.SET,030,000  ;SET CHARACTERISTICS FUNCTION
                        ;
                        ; GENERAL DEVICE HANDLER CODES
                        ;
                                .WORD.  IO.WLB,000,001  ;WRITE LOGICAL BLOCK
                                .WORD.  IO.RLB,000,002  ;READ LOGICAL BLOCK
                                .WORD.  IO.LOV,010,002  ;LOAD OVERLAY (DISK DRIVER)
                                .WORD.  IO.ATT,000,003  ;ATTACH A DEVICE TO A TASK
                                .WORD.  IO.DET,000,004  ;DETACH A DEVICE FROM A TASK
                        ;
                        ; DIRECTORY PRIMITIVE CODES
                        ;
                                .WORD.  IO.FNA,000,011  ;FIND FILE NAME IN DIRECTORY
                                .WORD.  IO.RNA,000,013  ;REMOVE FILE NAME FROM DIRECTORY
                                .WORD.  IO.ENA,000,014  ;ENTER FILE NAME IN DIRECTORY
                        ;
                        ; FILE PRIMITIVE CODES
                        ;
                                .WORD.  IO.CLN,000,007  ;CLOSE OUT LUN
                                .WORD.  IO.ULK,000,012  ;UNLOCK BLOCK
                                .WORD.  IO.ACR,000,015  ;ACCESS FOR READ
                                .WORD.  IO.ACW,000,016  ;ACCESS FOR WRITE
                                .WORD.  IO.ACE,000,017  ;ACCESS FOR EXTEND
                                .WORD.  IO.DAC,000,020  ;DE-ACCESS FILE
                                .WORD.  IO.RVB,000,021  ;READ VIRITUAL BLOCK
                                .WORD.  IO.WVB,000,022  ;WRITE VIRITUAL BLOCK
                                .WORD.  IO.EXT,000,023  ;EXTEND FILE
                                .WORD.  IO.CRE,000,024  ;CREATE FILE
                                .WORD.  IO.DEL,000,025  ;DELETE FILE
                                .WORD.  IO.RAT,000,026  ;READ FILE ATTRIBUTES
                                .WORD.  IO.WAT,000,027  ;WRITE FILE ATTRIBUTES
```

```
                                        .WORD.   IO.APV,010,030  ;PRIVILEGED ACP CONTROL
                                        .WORD.   IO.APC,000,030  ;ACP CONTROL
                                    :
                                    :
                                        .MACRO   FILIO$  A
                                        .ENDM    FILIO$
                                        .ENDM    FILIO$
```

```
                        ;
                        ; DEFINE THE I/O FUNCTION CODES THAT ARE SPECIFIC TO INDIVIDUAL DEVICES
                        ;
                                .MACRO  SPCIO$  $$$GBL
                                .MCALL  .WORD.,DEFIN$
                                .IF     IDN,<$$$GBL>,<DEF$G>
                                ...GBL=1
                                .IFF
                                ...GBL=0
                                .ENDC
                        ;
                        ; I/O FUNCTION CODES FOR SPECIFIC DEVICE DEPENDENT FUNCTIONS
                        ;
                                .WORD.  IO.WLV,100,001  ;(DECTAPE) WRITE LOGICAL REVERSE
                                .WORD.  IO.WLS,010,001  ;(COMM.) WRITE PRECEDED BY SYNC TRAIN
                                .WORD.  IO.WNS,020,001  ;(COMM.) WRITE, NO SYNC TRAIN
                                .WORD.  IO.WAL,010,001  ;(TTY) WRITE PASSING ALL CHARACTERS
                                .WORD.  IO.WMS,020,001  ;(TTY) WRITE SUPPRESSIBLE MESSAGE
                                .WORD.  IO.CCO,040,001  ;(TTY) WRITE WITH CANCEL CONTROL-O
                                .WORD.  IO.WBT,100,001  ;(TTY) WRITE WITH BREAKTHROUGH
                                .WORD.  IO.WLT,010,001  ;(DISK) WRITE LAST TRACK ON RK06
                                .WORD.  IO.WLC,020,001  ;(DISK) WRITE LOGICAL W/ WRITECHECK
                                .WORD.  IO.WPB,040,001  ;(RX11 DISK) WRITE PHYSICAL BLOCK
                                .WORD.  IO.WDD,044,001  ;(RX11 DISK) WRITE PHYSICAL W/ DELETED DATA
                                .WORD.  IO.RLV,100,002  ;(MAGTAPE,DECTAPE) READ REVERSE
                                .WORD.  IO.RST,001,002  ;(TTY) READ WITH SPECIAL TERMINATOR
                                .WORD.  IO.RAL,010,002  ;(TTY) READ PASSING ALL CHARACTERS
                                .WORD.  IO.RNE,020,002  ;(TTY) READ WITHOUT ECHO
                                .WORD.  IO.RNC,040,002  ;(TTY) READ - NO LOWER CASE CONVERT
                                .WORD.  IO.RTM,200,002  ;(TTY) READ WITH TIME OUT
                                .WORD.  IO.RDB,200,002  ;(CARD READER) READ BINARY MODE
                                .WORD.  IO.RHD,010,002  ;(COMM.) READ, STRIP SYNC
                                .WORD.  IO.RNS,020,002  ;(COMM.) READ, DON'T STRIP SYNC
                                .WORD.  IO.CRC,040,002  ;(COMM.) READ, DON'T CLEAR CRC
                                .WORD.  IO.RPB,040,002  ;(RX11 DISK) READ PHYSICAL BLOCK
                                .WORD.  IO.ATA,010,003  ;(TTY) ATTACH WITH AST'S
                                .WORD.  IO.GTS,000,005  ;(TTY) GET TERMINAL SUPPORT CHARACTERISTICS
                                .WORD.  IO.R1C,000,005  ;(AFC,AD01,UDC) READ SINGLE CHANNEL
                                .WORD.  IO.INL,000,005  ;(COMM.) INITIALIZATION FUNCTION
                                .WORD.  IO.TRM,010,005  ;(COMM.) TERMINATION FUNCTION
                                .WORD.  IO.RWD,000,005  ;(MAGTAPE,DECTAPE) REWIND
                                .WORD.  IO.SPB,020,005  ;(MAGTAPE) SPACE 'N' BLOCKS
                                .WORD.  IO.SPF,040,005  ;(MAGTAPE) SPACE 'N' EOF MARKS
                                .WORD.  IO.STC,100,005  ;(MAGTAPE) SET CHARACTERISTIC
                                .WORD.  IO.SEC,120,005  ;(MAGTAPE) SENSE CHARACTERISTIC
                                .WORD.  IO.RWU,140,005  ;(MAGTAPE,DECTAPE) REWIND AND UNLOAD
                                .WORD.  IO.SMO,160,005  ;(MAGTAPE) MOUNT & SET CHARACTERISTICS
                                .WORD.  IO.HNG,000,006  ;(TTY) HANGUP DIAL-UP LINE
                                .WORD.  IO.RBC,000,006  ;READ MULTICHANNELS (BUFFER DEFINES CHANNELS)
                                .WORD.  IO.MOD,000,006  ;(COMM.) SETMODE FUNCTION FAMILY
                                .WORD.  IO.HDX,010,006  ;(COMM.) SET UNIT HALF DUPLEX
                                .WORD.  IO.FDX,020,006  ;(COMM.) SET UNIT FULL DUPLEX
                                .WORD.  IO.SYN,040,006  ;(COMM.) SPECIFY SYNC CHARACTER
                                .WORD.  IO.EOF,000,006  ;(MAGTAPE) WRITE EOF
                                .WORD.  IO.RTC,000,007  ;READ CHANNEL - TIME BASED
                                .WORD.  IO.SAO,000,010  ;(UDC) SINGLE CHANNEL ANALOG OUTPUT
                                .WORD.  IO.SSO,000,011  ;(UDC) SINGLE SHOT, SINGLE POINT
```

```
                                .WORD.  IO.RPR,000,011  ;(TTY) READ WITH PROMPT
                                .WORD.  IO.MSO,000,012  ;(UDC) SINGLE SHOT, MULTI-POINT
                                .WORD.  IO.SLO,000,013  ;(UDC) LATCHING, SINGLE POINT
                                .WORD.  IO.MLO,000,014  ;(UDC) LATCHING, MULTI-POINT
                                .WORD.  IO.LED,000,024  ;(LPS11) WRITE LED DISPLAY LIGHTS
                                .WORD.  IO.SDO,000,025  ;(LPS11) WRITE DIGITAL OUTPUT REGISTER
                                .WORD.  IO.SDI,000,026  ;(LPS11) READ DIGITAL INPUT REGISTER
                                .WORD.  IO.SCS,000,026  ;(UDC) CONTACT SENSE, SINGLE POINT
                                .WORD.  IO.REL,000,027  ;(LPS11) WRITE RELAY
                                .WORD.  IO.MCS,000,027  ;(UDC) CONTACT SENSE, MULTI-POINT
                                .WORD.  IO.ADS,000,030  ;(LPS11) SYNCHRONOUS A/D SAMPLING
                                .WORD.  IO.CCI,000,030  ;(UDC) CONTACT INT - CONNECT
                                .WORD.  IO.LOD,000,030  ;(LPA11) LOAD MICROCODE
                                .WORD.  IO.MDI,000,031  ;(LPS11) SYNCHRONOUS DIGITAL INPUT
                                .WORD.  IO.DCI,000,031  ;(UDC) CONTACT INT - DISCONNECT
                                .WORD.  IO.XMT,000,031  ;(COMM.) TRANSMIT SPECIFIED BLOCK WITH ACK
                                .WORD.  IO.XNA,010,031  ;(COMM.) TRANSMIT WITHOUT ACK
                                .WORD.  IO.INI,00C,031  ;(LPA11) INITIALIZE
                                .WORD.  IO.HIS,000,032  ;(LPS11) SYNCHRONOUS HISTOGRAM SAMPLING
                                .WORD.  IO.RCI,000,032  ;(UDC) CONTACT INT - READ
                                .WORD.  IO.RCV,000,032  ;(COMM.) RECEIVE DATA IN BUFFER SPECIFIED
                                .WORD.  IO.CLK,000,032  ;(LPA11) START CLOCK
                                .WORD.  IO.MDO,000,033  ;(LPS11) SYNCHRONOUS DIGITAL OUTPUT
                                .WORD.  IO.CTI,000,033  ;(UDC) TIMER - CONNECT
                                .WORD.  IO.CON,000,033  ;(COMM.) CONNECT FUNCTION
                                                        ;(VT11) - CONNECT TASK TO DISPLAY PROCESSOR
                                .WORD.  IO.CPR,010,033  ;(COMM.) CONNECT NO TIMEOUTS
                                .WORD.  IO.CAS,020,033  ;(COMM.) CONNECT WITH AST
                                .WORD.  IO.CRJ,040,033  ;(COMM.) CONNECT REJECT
                                .WORD.  IO CBO,110,033  ;(COMM.) BOOT CONNECT
                                .WORD.  IO.CTR,210,033  ;(COMM.) TRANSPARENT CONNECT
                                .WORD.  IO.GNI,010,035  ;(COMM.) GET NODE INFORMATION
                                .WORD.  IO.GLI,020,035  ;(COMM.) GET LINK INFORMATION
                                .WORD.  IO.GLC,030,035  ;(COMM.) GET LINK INFO CLEAR COUNTERS
                                .WORD.  IO.GRI,040,035  ;(COMM.) GET REMOTE NODE INFORMATION
                                .WORD.  IO.GRC,050,035  ;(COMM.) GET REMOTE NODE ERROR COUNTS
                                .WORD.  IO.GRN,060,035  ;(COMM.) GET REMOTE NODE NAME
                                .WORD.  IO.CSM,070,035  ;(COMM.) CHANGE SOLO MODE
                                .WORD.  IO.CIN,100,035  ;(COMM.) CHANGE CONNECTION INHIBIT
                                .WORD.  IO.SPW,110,035  ;(COMM.) SPECIFY NETWORK PASSWORD
                                .WORD.  IO.CPW,120,035  ;(COMM.) CHECK NETWORK PASSWORD.
                                .WORD.  IO.NLB,130,035  ;(COMM.) NSP LOOPBACK
                                .WORD.  IO.DLB,140,035  ;(COMM.) DDCMP LOOPBACK

                                .WORD.  IO.STA,000,033  ;(LPA11) START DATA TRANSFER
                                .WORD.  IO.DTI,000,034  ;(UDC) TIMER - DISCONNECT
                                .WORD.  IO.DIS,000,034  ;(COMM.) DISCONNECT FUNCTION
                                                        ;(VT11) - DISCONNECT TASK FROM DISPLAY PROCESSOR
                                .WORD.  IO.MDA,000,034  ;(LPS11) SYNCHRONOUS D/A OUTPUT
                                .WORD.  IO.RTI,000,035  ;(UDC) TIMER - READ
                                .WORD.  IO.CTL,000,035  ;(COMM.) NETWORK CONTROL FUNCTION
                                .WORD.  IO.STP,000,035  ;(LPS11,LPA11) STOP IN PROGRESS FUNCTION
                                                        ;(VT11) - STOP DISPLAY PROCESSOR
                                .WORD.  IO.CNT,000,036  ;(VT11) - CONTINUE DISPLAY PROCESSOR
                                .WORD.  IO.ITI,000,036  ;(UDC) TIMER - INITIALIZE
```

```
                        ;
                        ; ICS/ICR I/O FUNCTIONS
                        ;
                                .WORD.  IO.CTY,000,007  ;CONNECT TO TERMINAL INTERRUPTS
                                .WORD.  IO.DTY,000,015  ;DISCONNECT FROM TERMINAL INTERRUPTS
                                .WORD.  IO.LDI,000,016  ;LINK TO DIGITAL INTERRUPTS
                                .WORD.  IO.UDI,010,023  ;UNLINK FROM DIGITAL INTERRUPTS
                                .WORD.  IO.LTI,000,017  ;LINK TO COUNTER MODULE INTERRUPTS
                                .WORD.  IO.UTI,020,023  ;UNLINK FROM COUNTER MODULE INTERRUPTS
                                .WORD.  IO.LTY,000,020  ;LINK TO REMOTE TERMINAL INTERRUPTS
                                .WORD.  IO.UTY,030,023  ;UNLINK FROM REMOTE TERMINAL INTERRUPTS
                                .WORD.  IO.LKE,000,024  ;LINK TO ERROR INTERRUPTS
                                .WORD.  IO.UER,040,023  ;UNLINK FROM ERROR INTERRUPTS
                                .WORD.  IO.NLK,000,023  ;UNLINK FROM ALL INTERRUPTS
                                .WORD.  IO.ONL,000,037  ;UNIT ONLINE
                                .WORD.  IO.FLN,000,025  ;UNIT OFFLINE
                                .WORD.  IO.RAD,000,021  ;READ ACTIVATING DATA

                        ;
                        ; IP11 I/O FUNCTIONS
                        ;
                                .WORD.  IO.MAO,010,007  ;MULTIPLE ANALOG OUTPUTS
                                .WORD.  IO.LEI,010,017  ;LINK EVENT FLAGS TO INTERRUPT
                                .WORD.  IO.RDD,010,020  ;READ DIGITAL DATA
                                .WORD.  IO.RMT,020,020  ;READ MAPPING TABLE
                                .WORD.  IO.LSI,000,022  ;LINK TO DSI INTERRUPTS
                                .WORD.  IO.UEI,050,023  ;UNLINK EVENT FLAGS
                                .WORD.  IO.USI,060,023  ;UNLINK FROM DSI INTERRUPTS
                                .WORD.  IO.CSI,000,026  ;CONNECT TO DSI INTERRUPTS
                                .WORD.  IO.DSI,000,027  ;DISCONNECT FROM DSI INTERRUPTS
                        ;
                        ;
                                .MACRO  SPCIO$  A
                                .ENDM   SPCIO$
                                .ENDM   SPCIO$
```

```
                        ;
                        ; DEFINE THE I/O CODES FOR USER-MODE DIAGNOSITCS.  ALL DIAGNOSTIC
                        ; FUNCTION ARE IMPLEMENTED AS A SUBFUNCTION OF I/O CODE 10 (OCTAL).
                        ;

                                .MACRO  UMDIO$  $$$GBL
                                .MCALL  .WORD.,DEFIN$
                                .IF IDN <$$$GBL>,<DEF$G>
                        ...GBL=1
                                .IFF
                        ...GBL=0
                                .ENDC

                        ;
                        ; DEFINE THE GENERAL USER-MODE I/O QUALIFIER BIT.
                        ;

                                .WORD.  IQ.UMD,004,000  ;USER MODE DIAGNOSTIC REQUEST

                        ;
                        ; DEFINE USER-MODE DIAGNOSTIC FUNCTIONS.
                        ;

                                .WORD.  IO.HMS,000,010  ;(DISK) HOME SEEK OR RECALIBRATE
                                .WORD.  IO.BLS,010,010  ;(DISK) BLOCK SEEK
                                .WORD.  IO.OFF,020,010  ;(DISK) OFFSET POSITION
                                .WORD.  IO.RDH,030,010  ;(DISK) READ DISK HEADER
                                .WORD.  IO.WDH,040,010  ;(DISK) WRITE DISK HEADER
                                .WORD.  IO.WCK,050,010  ;(DISK) WRITECHECK (NON-TRANSFER)
                                .WORD.  IO.RNF,060,010  ;(DECTAPE) READ BLOCK NUMBER FORWARD
                                .WORD.  IO.RNR,070,010  ;(DECTAPE) READ BLOCK NUMBER REVERSE
                                .WORD.  IO.LPC,100,010  ;(MAGTAPE) READ LONGITUDINAL PARITY CHAR
                                .WORD.  IO.ERS,110,010  ;(MAGTAPE) ERASE TAPE

                        ;
                        ; MACRO REDEFINITION TO NULL
                        ;

                                .MACRO  UMDIO$  A
                                .ENDM

                                .ENDM   UMDIO$
```

```
                                ;
                                ; HANDLER ERROR CODES RETURNED IN I/O STATUS BLOCK ARE DEFINED THROUGH THIS
                                ; MACRO WHICH THEN CONDITIONALLY INVOKES THE MESSAGE GENERATING MACRO
                                ; FOR THE QIOSYM.MSG FILE
                                ;
                                        .MACRO  .IOER.  SYM,LO,MSG
                                        DEFIN$  SYM,LO
                                        .IF     GT,$$MSG
                                        .MCALL  .IOMG.
                                        .IOMG.  SYM,LO,<MSG>
                                        .ENDC
                                        .ENDM   .IOER.
                                ;
                                ; I/O ERROR CODES ARE DEFINED THOUGH THIS MACRO WHICH THEN INVOKES THE
                                ; ERROR MESSAGE GENERATING MACRO, ERROR CODES -129 THROUGH -256
                                ; ARE USED IN THE QIOSYM.MSG FILE
                                ;
                                        .MACRO  .QIOE.  SYM,LO,MSG
                                        DEFIN$  SYM,LO
                                        .IF     GT,$$MSG
                                        .MCALL  .IOMG.
                                        .IOMG.  SYM,<LO-128.>,<MSG>
                                        .ENDC
                                        .ENDM   .QIOE.
                                ;
                                ; CONDITIONALLY GENERATE DATA FOR WRITING A MESSAGE FILE
                                ;
                                        .MACRO  .IOMG.  SYM,LO,MSG
                                        .WORD   -^O<LO>
                                        .ASCIZ  ^MSG^
                                        .EVEN
                                        .IIF    LT,^O<$$$MAX+<LO>>,$$$MAX=-^O<LO>
                                        .ENDM   .IOMG.
                                ;
                                ; DEFINE THE SYMBOL SYM WHERE LO IS IS THE LOW ORDER BYTE, HI IS THE HIGH BYTE
                                ;
                                        .MACRO  .WORD.  SYM,LO,HI
                                        DEFIN$  SYM,<HI*400+LO>
                                        .ENDM   .WORD.
```

```
                                .TITLE  ERRMSG
                                .IDENT  \V02.03\

000000                                  .PSECT  RSXTX0, RO, D, GBL
000000                                  .PSECT  RSXTX1, RO, D, GBL

000000                                  .PSECT  RSXTB0, RO, D, GBL
000000                          RSXER0::
        000100                          .REPT   64.
                                        .WORD   BADRR
                                .ENDR

000000                                  .PSECT  RSXTB1, RO, D, GBL
000000                          RSXER1::
        000100                          .REPT   64.
                                        .WORD   BADRR
                                .ENDR

        000001                  $$MSG=  1

                                .MACRO  .IOMG.  SYM,    LO,     MSG
                                .IF GE LO+64.
                                    .PSECT  RSXTX0
                                .IFF
                                    .PSECT  RSXTX1
                                .ENDC
                                .LIST
                          $$$$$$= .
                                .NLIST
                                .IRPC   CHR,    <MSG>
                                    CHR1=       ''CHR
                                    .IIF EQ ''CHR-'(,   CHR1=   '[
                                    .IIF EQ ''CHR-'),   CHR1=   ']
                                    .IIF EQ ''CHR-'',   CHR1=   'K
                                    .IIF EQ ''CHR-'-,   CHR1=   '_
                                    .IF  LE ''CHR-'Z
                                        .IF GE ''CHR-'A
                                            CHR1=       ''CHR+40
                                        .ENDC
                                    .ENDC
                                    .BYTE       CHR1
                                .ENDM
                                .BYTE   0                   ;ASCIZ-TYPE STRING
                                .IF GE LO+64.
                                    .PSECT  RSXTB0
                          .=    RSXER0+<<LO+64.>*2>
                                .IFF
                                    .PSECT  RSXTB1
                          .=    RSXER1+<<LO+128.>*2>
                                .ENDC
                                .WORD   $$$$$$
                                .ENDM   .IOMG.

000200                          IOERR$
        000000'                 $$$$$$= .
        000017'                 $$$$$$= .
        000045'                 $$$$$$= .
```

```
000066'                 $$$$$$= .
000115'                 $$$$$$= .
000151'                 $$$$$$= .
000175'                 $$$$$$= .
000221'                 $$$$$$= .
000251'                 $$$$$$= .
000277'                 $$$$$$= .
000324'                 $$$$$$= .
000353'                 $$$$$$= .
000412'                 $$$$$$= .
000427'                 $$$$$$= .
000454'                 $$$$$$= .
000477'                 $$$$$$= .
000523'                 $$$$$$= .
000554'                 $$$$$$= .
000604'                 $$$$$$= .
000650'                 $$$$$$= .
000707'                 $$$$$$= .
000734'                 $$$$$$= .
000756'                 $$$$$$= .
001002'                 $$$$$$= .
001046'                 $$$$$$= .
001105'                 $$$$$$= .
000000'                 $$$$$$= .
000020'                 $$$$$$= .
001132'                 $$$$$$= .
001163'                 $$$$$$= .
001177'                 $$$$$$= .
001217'                 $$$$$$= .
001234'                 $$$$$$= .
001272'                 $$$$$$= .
001313'                 $$$$$$= .
001336'                 $$$$$$= .
001373'                 $$$$$$= .
001437'                 $$$$$$= .
001500'                 $$$$$$= .
001542'                 $$$$$$= .
001577'                 $$$$$$= .
001632'                 $$$$$$= .
001671'                 $$$$$$= .
001721'                 $$$$$$= .
001756'                 $$$$$$= .
002023'                 $$$$$$= .
002046'                 $$$$$$= .
000063'                 $$$$$$= .
000124'                 $$$$$$= .
000144'                 $$$$$$= .
000167'                 $$$$$$= .
000204'                 $$$$$$= .
002066'                 $$$$$$= .
002140'                 $$$$$$= .
002164'                 $$$$$$= .
002234'                 $$$$$$= .
002307'                 $$$$$$= .
002327'                 $$$$$$= .
002366'                 $$$$$$= .
002431'                 $$$$$$= .
```

```
002473'                 $$$$$$= .
002530'                 $$$$$$= .
002576'                 $$$$$$= .
002621'                 $$$$$$= .
002656'                 $$$$$$= .
002703'                 $$$$$$= .
002725'                 $$$$$$= .
002743'                 $$$$$$= .
002763'                 $$$$$$= .
003015'                 $$$$$$= .
000230'                 $$$$$$= .
000267'                 $$$$$$= .
000323'                 $$$$$$= .
000340'                 $$$$$$= .
000365'                 $$$$$$= .
000410'                 $$$$$$= .
000446'                 $$$$$$= .
000501'                 $$$$$$= .
000525'                 $$$$$$= .
000550'                 $$$$$$= .
000572'                 $$$$$$= .
000652'                 $$$$$$= .
000707'                 $$$$$$= .
000766'                 $$$$$$= .
001016'                 $$$$$$= .
```

```
                                        .PSECT  RSXTX2, RO, D, GBL
                                        .PSECT  RSXTX3, RO, D, GBL

                                        .PSECT  RSXTB2, RO, D, GBL
                                RSXER2::
                        000100          .REPT   64.
                                            .WORD   BADRR
                                        .ENDR

                                        .PSECT  RSXTB3, RO, D, GBL
                                RSXER3::
                        000100          .REPT   64.
                                            .WORD   BADRR
                                        .ENDR

                        000001          $$MSG=  1

                                        .MACRO  .IOMG.  SYM,    LO,     MSG
                                        .IF GE LO+192.
                                            .PSECT  RSXTX2
                                        .IFF
                                            .PSECT  RSXTX3
                                        .ENDC
                                        .LIST
                                $$$$$$= .
                                        .NLIST
                                        .IRPC   CHR,    <MSG>
                                            CHR1=       ''CHR
                                            .IIF EQ ''CHR-'(,   CHR1=   ':
                                            .IIF EQ ''CHR-'),   CHR1=   ';
                                            .IIF EQ ''CHR-'',   CHR1=   'K
                                            .IIF EQ ''CHR-'-,   CHR1=   '_
                                            .IF  LE ''CHR-'Z
                                                .IF GE ''CHR-'A
                                                    CHR1=       ''CHR+40
                                                .ENDC
                                            .ENDC
                                            .BYTE       CHR1
                                        .ENDM
                                        .BYTE   0                   ;ASCIZ-TYPE STRING
                                        .IF GE LO+192.
                                            .PSECT  RSXTB2
                                .=      RSXER2+<<LO+192.>*2>
                                        .IFF
                                            .PSECT  RSXTB3
                                .=      RSXER3+<<LO+256.>*2>
                                        .ENDC
                                        .WORD   $$$$$$
                                        .ENDM   .IOMG.

                                        DRERR$
                        000000'         $$$$$$= .
                        000035'         $$$$$$= .
                        000072'         $$$$$$= .
                        000127'         $$$$$$= .
                        000175'         $$$$$$= .
                        000215'         $$$$$$= .
```

```
                000251'                 $$$$$$= .
                000271'                 $$$$$$= .
                000340'                 $$$$$$= .
                000373'                 $$$$$$= .
                000433'                 $$$$$$= .
                000462'                 $$$$$$= .
                000516'                 $$$$$$= .
                000542'                 $$$$$$= .
                000562'                 $$$$$$= .
                000612'                 $$$$$$= .
                000000'                 $$$$$$= .
                000045'                 $$$$$$= .
                000077'                 $$$$$$= .
                000132'                 $$$$$$= .
                000152'                 $$$$$$= .
                000215'                 $$$$$$= .
                000237'                 $$$$$$= .
                000271'                 $$$$$$= .
                000316'                 $$$$$$= .
                000364'                 $$$$$$= .
                000406'                 $$$$$$= .
                000422'                 $$$$$$= .
                000451'                 $$$$$$= .
                000501'                 $$$$$$= .
                000540'                 $$$$$$= .
                000576'                 $$$$$$= .
                000612'                 $$$$$$= .
                000651'                 $$$$$$= .
                000711'                 $$$$$$= .

107
108             000001                          .END
```

ERRMSG MACRO M1113 08-FEB-80 11:59 PAGE 10-2
SYMBOL TABLE

```
BADRR = ****** GX        IE.DAO= 177763        IE.ISQ= 177703        IE.PES= 177655        IE.WCK= 177652
CHR1  = 000144           IE.DFU= 177750        IE.ITI= 177643        IE.PNS= 177642        IE.WER= 177737
IE.ABO= 177761           IE.DNA= 177771        IE.ITP= 177650        IE.PRI= 177760        IE.WLK= 177764
IE.ACT= 177771           IE.DNR= 177775        IE.ITS= 177770        IE.PTS= 177775        IE.WOV= 177653
IE.ADP= 177636           IE.DUN= 177767        IE.IUI= 177645        IE.RAC= 177724        IE.2DV= 177720
IE.ALC= 177654           IE.DUP= 177707        IE.LCK= 177745        IE.RAT= 177723        IS.BV = 000005
IE.ALG= 177654           IE.EOF= 177766        IE.LNL= 177646        IE.RBG= 177730        IS.CC = 001401
IE.ALN= 177736           IE.EOT= 177702        IE.MAP= 177657        IE.RBS= 177761        IS.CLR= 000000
IE.AST= 177660           IE.EOV= 177765        IE.MOD= 177753        IE.RCN= 177722        IS.CR = 006401
IE.BAD= 177777           IE.EXP= 177665        IE.NBF= 177731        IE.RER= 177740        IS.EOT= 002001
IE.BBE= 177710           IE.FEX= 177717        IE.NBK= 177727        IE.RNM= 177715        IS.ESC= 015401
IE.BCC= 177676           IE.FHE= 177705        IE.NDR= 177670        IE.RSU= 177757        IS.ESQ= 115401
IE.BDI= 177714           IE.FIX= 177767        IE.NFI= 177704        IE.SDP= 177635        IS.PES= 100001
IE.BDR= 177716           IE.FLN= 177657        IE.NFW= 177673        IE.SNC= 177735        IS.PND= 000000
IE.BDV= 177711           IE.FOP= 177713        IE.NLK= 177661        IE.SPC= 177772        IS.RDD= 000002
IE.BHD= 177700           IE.HFU= 177744        IE.NLN= 177733        IE.SQC= 177734        IS.SET= 000002
IE.BLB= 177672           IE.HWR= 177772        IE.NNC= 177663        IE.SRE= 177762        IS.SPD= 000002
IE.BLK= 177754           IE.IBS= 177647        IE.NNL= 177662        IE.STK= 177706        IS.SUC= 000001
IE.BNM= 177712           IE.IDU= 177644        IE.NNN= 177674        IE.TCH= 177765        IS.TAB= 004401
IE.BTF= 177664           IE.IEF= 177637        IE.NOD= 177751        IE.TMM= 177671        IS.TMO= 000002
IE.BTP= 177725           IE.IES= 177656        IE.NSF= 177746        IE.TMO= 177666        QI.VER= 000331
IE.BVR= 177701           IE.IFC= 177776        IE.NST= 177660        IE.ULK= 177653        RSXER0  000000RG      004
IE.BYT= 177755           IE.IFU= 177747        IE.NSW= 177756        IE.ULN= 177773        RSXER1  000000RG      005
IE.CKP= 177766           IE.ILL= 177726        IE.NVR= 177652        IE.UNS= 177774        RSXER2  000000RG      010
IE.CKS= 177742           IE.ILU= 177640        IE.NVW= 177651        IE.UPN= 177777        RSXER3  000000RG      011
IE.CLO= 177732           IE.ILV= 177755        IE.OFL= 177677        IE.VER= 177774        $$MSG = 000001
IE.CNR= 177667           IE.INS= 177776        IE.ONP= 177773        IE.WAC= 177743        $$$$$$= 000711R       007
IE.CON= 177752           IE.IOP= 177655        IE.OVR= 177756        IE.WAT= 177741        ...GBL= 000000
IE.DAA= 177770           IE.IPR= 177641

. ABS.  000000      000
        000000      001
RSXTX0  003052      002
RSXTX1  001046      003
RSXTB0  000200      004
RSXTB1  000200      005
RSXTX2  000643      006
RSXTX3  000741      007
RSXTB2  000200      010
RSXTB3  000200      011
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  11739 WORDS  ( 46 PAGES)
DYNAMIC MEMORY:  12366 WORDS  ( 47 PAGES)
ELAPSED TIME:  00:07:16
ERRMSG,DM1:[305,301]ERRMSG/-SP=QIOMAC,IOERR
```

FILEID**ERROR

```
EEEEEEEEEE   RRRRRRRR     RRRRRRRR       000000     RRRRRRRR
EEEEEEEEEE   RRRRRRRR     RRRRRRRR       000000     RRRRRRRR
EE           RR      RR   RR      RR   00      00   RR      RR
EE           RR      RR   RR      RR   00      00   RR      RR
EE           RR      RR   RR      RR   00      00   RR      RR
EE           RR      RR   RR      RR   00      00   RR      RR
EEEEEEEE     RRRRRRRR     RRRRRRRR     00      00   RRRRRRRR
EEEEEEEE     RRRRRRRR     RRRRRRRR     00      00   RRRRRRRR
EE           RR  RR       RR  RR       00      00   RR  RR
EE           RR  RR       RR  RR       00      00   RR  RR
EE           RR    RR     RR    RR     00      00   RR    RR        ....
EE           RR    RR     RR    RR     00      00   RR    RR        ....
EEEEEEEEEE   RR      RR   RR      RR     000000     RR      RR      ....
EEEEEEEEEE   RR      RR   RR      RR     000000     RR      RR      ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:06
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    ERROR            HANDLE ERRORS
```

ERROR MACRO M1113 08-FEB-80 12:06 PAGE 1

```
     1          000000                RT11    =  0          ;NOT RT-11
     2          000000                RSTS    =  0          ;NOT RSTS/E
     3          000115                RSX     = 'M          ;RSX-11/M
     4
     5          000000                EIS     =  0          ;NO EIS
     6          000000                FIS     =  0          ;NO FIS
     7          000000                FPP     =  0          ;NO FPP
     8
     9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
    10
    11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
ERROR   MACRO M1113  08-FEB-80 12:06  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  ERROR
                                .IDENT  \V02.03\

                                .SBTTL  ERROR           HANDLE ERRORS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       Check if an error happened while processing an error.  If so, just
                                ;       die (return user to monitor).  If not, branch to error processor
                                ;       in overlay.
                                ;
                                ; CALLING SEQUENCE:
                                ;       TRAP    nnn
                                ;
                                ; INPUTS:
                                ;       The number of the issued emt
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       EXITIT  - An ERROR in ERROR
                                ;       XERROR  - The error routine in overlay
                                ;
                                ; SIDE EFFECTS:
                                ;       A WHOLE BUNCH OF THEM!
                                ;--
000000                                  .PSECT  CODE

                                ;
                                ;FIRST CHECK IF ERROR ALREADY IN PROGRESS.  IF SO THEN DIE.....
                                ;
000000                          TERROR::
000000  006216                          ASR     (SP)            ;RSX MULTIPLIES TRAP BY 2
000002                          ERROR::
000002  005737  004770'                 TST     ERRPTR          ;ERROR ALREADY IN PROGRESS?
000006  001402                          BEQ     1$              ;NO, GOOD
000010  000137  000000G                 JMP     DIE             ;YES, WE MUST USE THIS LAST RESORT

000014                          1$:     LDFPS   FPMODE          ;RESET PROPER FLOATING POINT STATUS
000024  042737  100000  000000'        BIC     #SY.BRK,SYSTAT  ;RESET FROM ^C
                                ;+
                                ; IF IN DEBUG MODE SAVE ALL OF THE REGISTERS.
                                ;-
000032  016601  000002                  MOV     2(SP),R1        ;POINTER PAST INSTRUCTION
000036  005741                          TST     -(R1)           ;BACK UP 1 INSTRUCTION
000040  010137  004770'                 MOV     R1,ERRPTR       ;AND SAVE POINTER TO TRAP
000044                                  POP     R1              ;Get the Error Number
000046  010137  004772'                 MOV     R1,ERRNUM       ;SAVE ERROR NUMBER

                                ;+
                                ; CALL ERROR HANDLER IN OVERLAY
                                ;-
000052  000137  000000G                 JMP     XERROR          ;BRANCH TO ERROR PROCESSOR

                                ;+
                                ; HERE IS WHERE SOME BAD ERRORS COME TO WHEN THEY HAPPEN AS A
                                ; SUBROUTINE CALL OR RETURN VALUE FOR A FRAME
```

```
ERROR   MACRO M1113  08-FEB-80 12:06  PAGE 44-1
ERROR           HANDLE ERRORS

     88                                 ;-
     89 000056  052737  000001  000000' SYSER:: BIS     #SY.BAD,SYSTAT  ;SET POSSIBILITY OF A LOUSY ERROR
     90 000064  104402                          SYSERR
     91 000066  104432                  NYIER:: NYIERR
     92 000070  104407                  SYNER:: SYNERR
     93 000072  104417                  DOMER:: DOMERR
     94
     96                                 ;+
     97                                 ; TABLE FOR RSX ERRORS
     98                                 ;-
     99 000000                                  .PSECT  TABLES
    100                                         .EVEN
    101
    102 000000                          SSTVT::                         ;SST VECTOR TABLE
    103 000000  000056'                         .WORD   SYSER           ;ODD OR NON-EXISTENT MEMORY
    104 000002  000000                          .WORD   0               ;MEMORY PROTECT ERROR
    105 000004  000056'                         .WORD   SYSER           ;T-BIT OR BPT
    106 000006  000056'                         .WORD   SYSER           ;IOT INSTRUCTION
    107 000010  000056'                         .WORD   SYSER           ;RESERVED INSTRUCTION
    108 000012  000000'                         .WORD   TERROR          ;BAD EMT
    109 000014  000000'                         .WORD   TERROR          ;TRAP INSTRUCTION
    113 000016  000056'                         .WORD   SYSER           ;WE DON'T HAVE FIS
```

F 5

```
ERROR   MACRO M1113  08-FEB-80 12:06  PAGE 47
ERROR           HANDLE ERRORS

      1         000001                          .END
```

G 5

ERROR MACRO M1113 08-FEB-80 12:06 PAGE 47-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073   C.BLSZ= 000140   D.DEAL= 000003   EDIT  = 000003
AC1    =%000001          CHRLPR= 000072   C.BUFF= 177760   D.DECD= 000002   EIS   = 000000
AC2    =%000002          CHRMNS= 000137   C.BUFL= 177762   D.DIV = 000000   ELESIZ 005132R     025
AC3    =%000003          CHRMUL= 000075   C.CCNT= 177766   D.DMD = 000010   ENDCOR 000004R     025
AC4    =%000004          CHRNEG= 000100   C.CPTR= 177764   D.DQAD= 177777   EOF   = 104475
AC5    =%000005          CHRPDL= 000271   C.CZ  = 001000   D.DROP= 000002   ERRMAX= 000166
ALLINT  003274R     025  CHRPLU= 000055   C.DIG = 000040   D.ENCD= 000002   ERRNUM 004772R     025
ARYRET  003276R     025  CHRPWR= 000120   C.FASW= 020000   D.EQ  = 000000   ERROR  000002RG    020
ASCQUO= 000042           CHRQST= 000121   C.FLEN= 000010   D.EXPD= 000002   ERRPTR 004770R     025
A.ALEN= 000004           CHRQUO= 000113   C.FLSH= 004000   D.FGET= 000005   ERRO   004774R     025
A.ARG = 000040           CHRRBR= 000047   C.FRC = 000001   D.FMT = 000004   EXEBUF 005054R     025
A.ASB = 000200           CHRRPR= 000042   C.IDEN= 000010   D.FPNT= 000005   E$CT   005056R     025
A.BITS= 000003           CHRSEM= 000074   C.NAME= 000102   D.FPUT= 000005   E$PT   005060R     025
A.CON = 000020           CHRSMO= 000112   C.NOBF= 000400   D.GE  = 000000   FA.APD= 000100
A.DIMS= 000002           CHRSPA= 000040   C.NUM = 000020   D.GT  = 000000   FA.CRE= 000010
A.DIM1= 000006           CHRUPA= 000131   C.OP  = 000004   D.IBM = 000002   FA.DLK= 001000
A.NUM = 000006           CHR.BS= 000010   C.OPN = 010000   D.IND = 000002   FA.ENB= 100000
A.PTR = 000000           CHR.CC= 000003   C.RDLN= 177776   D.INNR= 000013   FA.EXC= 002000
A.TMP = 000010           CHR.CR= 000015   C.REWT= 002000   D.INTR= 000006   FA.EXT= 000004
BADDIR= 104463           CHR.CZ= 000032   C.RND = 000010   D.LE  = 000000   FA.NSP= 000100
BADERR= 104437           CHR.GT= 000046   C.SBIT= 177756   D.LOG = 000001   FA.NSR= 000200
BLZBUB  005116R     025  CHR.LF= 000012   C.SPFX= 000100   D.LT  = 000000   FA.POS= 010000
BRSELF  005046R     025  CHR.LT= 000043   C.SPGL= 000200   D.MAX = 000000   FA.RD = 000001
CCNT    005072R     025  CHR.SI= 000017   C.SPLX= 000001   D.MEM = 000002   FA.RWD= 004000
CHANEX  003060R     025  CHR.SL= 000057   C.SPRS= 000002   D.MIN = 000000   FA.SEQ= 040000
CHAN00  000256R     025  CHR.SO= 000016   C.SRC = 177774   D.MUL = 000000   FA.SHR= 000040
CHAN01  000330R     025  CHR.TB= 000011   C.SRCP= 000024   D.NE  = 000000   FA.TMP= 000020
CHAN02  000512R     025  CHR.UL= 000106   C.TT  = 000004   D.NN  = 000000   FA.WCK= 020000
CHAN03  000674R     025  CHR..A= 000141   C.WIDC= 177770   D.NR  = 000000   FA.WRT= 000002
CHAN04  001056R     025  CHR..E= 000145   C.WIDR= 177772   D.OR  = 000000   FD.BLK= 000010
CHAN05  001240R     025  CHR..S= 000163   DEBUG = 000000   D.PLUS= 000000   FD.CCL= 000002
CHAN06  001422R     025  CHR..T= 000164   DEFERR= 104405   D.PSUB= 000006   FD.COM= 020000
CHAN07  001604R     025  CHR..U= 000165   DEVIL   005114R     025 D.PWR = 000001   FD.CR = 000002
CHAN08  001766R     025  CHR..Z= 000172   DEVNTF= 104470   D.QQAD= 000000   FD.DIR= 000010
CHAN09  002150R     025  CHR..0= 000060   DIE   = ****** GX D.QUAD= 000000   FD.FTN= 000001
CHAN10  002332R     025  CHR..1= 000061   DIGITS  000006R     025 D.RES = 000000   FD.F11= 040000
CHAN11  002514R     025  CH.AND= 000001   DIMAX = 000020   D.RHO = 000000   FD.INS= 000010
CHAN12  002676R     025  CMDBUF  003250R     025 DIMBLK  003112R     025 D.ROT = 000003   FD.ISP= 002000
CHAN13  000022R     023  CMDMAX= 000010   DIMPTR  003114R     025 D.SCON= 000006   FD.MNT= 100000
CHAN14  000204R     023  CNTSIZ  005130R     025 DIMTYP  003116R     025 D.SDIF= 000006   FD.OSP= 004000
CHNOPN= 104471           COMBIN  005076R     025 DIRFUL= 104466   D.SUB = 000000   FD.PLC= 000004
CHRAND= 000051           COMCNT  005126R     025 DOMER   000072RG    020 D.SUBS= 000006   FD.PRN= 000004
CHRBEL= 000007           COMERR= 104425   DOMERR= 104417   D.TAKE= 000002   FD.PSE= 010000
CHRBKR= 000133           COMPTR  005122R     025 DPTERR= 104416   D.TRAN= 000004   FD.RAH= 000001
CHRBOL= 000200           COMRTN  003156R     025 DYDCNT  003224R     025 D.UNON= 000006   FD.RAN= 000002
CHRBOX= 000114           COMSKP  005124R     025 D.AND = 000000   D.$MAX= 000000   FD.REC= 000001
CHRCMA= 000054           CORPTR  005120R     025 D.BXCT= 000007   D.$MIN= 000000   FD.RWM= 000001
CHRCOL= 000076           CS.ASS= 000200   D.BXIO= 000007   D.$MUL= 000000   FD.SDI= 000020
CHRDEL= 000107           CS.BOO= 000010   D.BXNL= 000007   D.$PLU= 000000   FD.SQD= 000040
CHRDIV= 000053           CS.FLO= 000020   D.BXPP= 000007   D.$RES= 000000   FD.TTY= 000004
CHRDLR= 000176           CS.IDN= 000040   D.BXPW= 000007   D.$SUB= 000000   FD.WBH= 000002
CHRDLT= 000110           CS.ONE= 000100   D.BXRL= 000007   D.1COM= 000001   FF.CHR= 000005
CHRDOT= 000056           CTEPS   000152R     025 D.CIRC= 000001   D.1EXP= 000002   FF.NV = 000003
CHREP = 000105           CTMAX   000212R     025 D.COMB= 000001   D.1ROT= 000003   FF.POE= 000002
CHRERR= 104431           CTRMAX  000172R     025 D.COMP= 000001   EDICHR  005063R     025 FF.RWD= 000001
CHRFIL= 000177           C.ALOC= 000162   D.CONT= 000006   EDICNT  005064R     025 FF.RWF= 000006
CHRFRO= 000135           C.BLOK= 000064   D.CTLM= 000000   EDIFLG  005062R     025 FF.SPC= 000004
```

ERROR MACRO M1113 08-FEB-80 12:06 PAGE 47-2
SYMBOL TABLE

```
FHEADR  005012R  025 F.FLB = 000100      HSHSIZ= 000200      MONLEN  003130R  025 M.ROLL= 000000
FHRET   005040R  025 F.FNAM= 000110      ICHAN   005050R  025 MONPTR  003122R  025 M.SATA= 000001
FILNTF= 104467       F.FNB = 000102      ILFNAM= 104464      MONSFT  003126R  025 M.SCAN= 000011
FIS   = 000000       F.FNDF= 000010      ILLCHN= 104540      MONTYP  003124R  025 M.SHAP= 000000
FISCON= 046000       F.FTYP= 000116      ILLSW = 104565      MONVAL  003134R  025 M.SIGN= 000000
FIXPTR  003074R  025 F.FUN = 000010      INDERR= 104410      M.ABS = 000000       M.SUBR= 000001
FLAGS   004764R  025 F.FVER= 000120      INNTMP  003250R  025 M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R  025 F.HEAD= 000006      INNTM2  005016R  025 M.BXAV= 000007      M.SYN = 000201
FLINE   005014R  025 F.HIBK= 000004      INPTR   000064R  025 M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R  025 F.LBL = 000020      INSNAM= 104462       M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R  025 F.LINE= 000012      IOERR = 104477       M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R  025 F.LOC = 000004      I.CNT = 177776       M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010       F.LSTC= 000004      I.DIM = 177772       M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R  025 F.LUN = 000042      I.LEN = 000012       M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R  025 F.MBCT= 000054      I.OERR= 000062       M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106       F.MBC1= 000055      I.OLD = 177770       M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002       F.MBFG= 000056      I.PTR = 177774       M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001       F.MON = 000002      I.STR = 177766       M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006       F.NIL = 000001      LABERR= 104406       M.BXRL= 000007      NAMBUF  003240R  025
FO.WRT= 000016       F.NRBD= 000024      LABLPT  005042R  025 M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R  025 F.NREC= 000030      LARGBL  003162R  025 M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000       F.OPN = 000010      LARGCV  003176R  025 M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004       F.OVBS= 000030      LARGFL  003160R  025 M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R  025 F.PREV= 000000      LARGLE  003174R  025 M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R  025 F.PRNT= 100000      LARGLN  003166R  025 M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R  025 F.PROT= 010000      LARGPT  003164R  025 M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100       F.RACC= 000016      LARGSF  003172R  025 M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200       F.RATT= 000001      LARGTY  003170R  025 M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400       F.RCNM= 000034      LARGVA  003200R  025 M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000       F.RCTL= 000017      LBINC   005100R  025 M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076       F.READ= 004000      LBUFF   003316R  025 M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040       F.REDF= 000040      LBUFLN= 000621       M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062       F.RES = 000040      LCOUNT  005106R  025 M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070       F.RET = 177772      LENERR= 104412       M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057       F.RSIZ= 000002      LENTAB  000054R  025 M.GDWN= 000003      ND.$ES= 000020
F.BITS= 177774       F.RTYP= 000000      LEXPTR  003100R  025 M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026       F.R2  = 000014      LFTOP   005066R  025 M.IBM = 000003      ND.$NS= 000040
F.BKDS= 000020       F.R3  = 000016      LINC    005102R  025 M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050       F.R4  = 000020      LINE    005016R  025 M.INSP= 000006      NEGSGN  005063R  025
F.BKP1= 000051       F.SEQN= 000100      LININC  005026R  025 M.IOTA= 000000      NINPT1  003302R  025
F.BKST= 000024       F.SPDV= 000072      LISIZE= 000020       M.LIT = 000205      NINPT2  003300R  025
F.BKVB= 000064       F.SPUN= 000074      LITOP   005072R  025 M.LN  = 000001      NINSTK  003304R  025
F.BR  = 000020       F.STBK= 000036      LNLNG = 104541       M.MMD = 000010      NINTMP  003310R  025
F.CHR = 000075       F.STK = 000022      LOCLCT  005036R  025 M.NEG = 000000      NINTYP  003306R  025
F.CNTG= 000034       F.STOP= 000004      LOWCOR  000002R  025 M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010       F.STP = 000001      LSTERP  004766R  025 M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046       F.SUSP= 000002      LWIDTH  000070R  025 M.NUM = 000204      NPRT1   004762R  025
F.DSPT= 000044       F.TRC = 000002      L.BITS= 000003       M.OUTR= 000012      NULL    007364R  025
F.DVNM= 000134       F.UNIT= 000136      L.CNT = 000002       M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010       F.URBD= 000020      L.PRNT= 000001       M.PRDS= 000006      NUMDFL  003230R  025
F.EFN = 000050       F.VAL = 177776      L.PTR = 000000       M.QQAD= 000000      NUMEFL  003233R  025
F.END = 002000       F.VBN = 000064      L.PTR1= 000004       M.QUAD= 000000      NUMESN  003236R  025
F.EOBB= 000032       F.VBSZ= 000060      L.SUBS= 000002       M.RAND= 000012      NUMFLG  003230R  025
F.ERR = 000052       GNOME   003102R  025 MACHIN= 000106       M.RAVL= 000000      NUMLEN  000062R  025
F.FACC= 000043       HASHC   005136R  025 MAXCHN= 000014       M.RCIP= 000000      NUMNEG  003232R  025
F.FFBY= 000014       HASHTB  005140R  025 MONBLK  003120R  025 M.RDCT= 000011      NUMQFL  003234R  025
F.FLAG= 000002       HSHMSK= 177601       MONCVT  003132R  025 M.REVS= 000003      NUMRIG  003231R  025
```

```
ERROR   MACRO M1113  08-FEB-80 12:06  PAGE 47-3
SYMBOL TABLE

NUMWID  003226R     025 RANERR= 104411          R.PREV= 000004          SY.BS = 000010          T.LIT = 000020
NUMZFL  003235R     025 RARGBL  003204R     025 R.PTR = 000000          SY.CLR= 000040          T.MON = 000010
NYIER   000066RG    020 RARGCV  003220R     025 R.SEQ = 000003          SY.ERP= 000020          T.NIL = 000004
NYIERR= 104432          RARGFL  003202R     025 R.VAR = 000002          SY.NOU= 000100          T.NUM = 002000
N.BLSZ= 000036          RARGLE  003216R     025 R6    =%000006          SY.OPN= 000002          T.OPFN= 000001
N.DESC= 000020          RARGLN  003210R     025 R7    =%000007          SY.PMT= 001000          T.OTHR= 040000
N.DEV = 000032          RARGPT  003206R     025 SI    =%000005          SY.SRT= 000400          T.SOP = 000002
N.DEVN= 000034          RARGSF  003214R     025 SIDAMN= 104433          SY.SVC= 000004          T.SPOP= 000040
N.DID = 000024          RARGTY  003212R     025 SIFRAM  003314R   025 SY.SVL= 000200          T.TMP = 000200
N.DVNM= 000032          RARGVA  003222R     025 SISTAK  006350R   025 S.AB  = 177776          T.VAL = 100000
N.EXT = 000014          RBASE   005074R     025 SITOP   006346R   025 S.ASB = 100000          T.VAR = 001000
N.FID = 000000          RCOUNR  005112R     025 SITP    007144R   025 S.BFHD= 000020          UPPER = 177400
N.FNAM= 000006          RCOUNT  005110R     025 SIWARN= 000200          S.BITS= 000002          UTNAME  000344R     023
N.FTYP= 000014          REDCNT  005070R     025 SIZTAB  000044R   025 S.CHR1= 000003          VALERR= 104413
N.FVER= 000016          REDIGN  005066R     025 SI.BIT= 177774          S.FATT= 000016          VARBLE  007400R     025
N.NAME= 000006          REDOFF  005072R     025 SI.LEN= 177772          S.FDB = 000140          VARBLS  000000R     025
N.NEXT= 000022          REENTR  003060R     025 SI.RET= 177772          S.FLB = 040000          VERNUM  000000R     025
N.PPN = 000024          RESBIT  003142R     025 SI.VAL= 177776          S.FNAM= 000006          VRSION= 000002
N.STAT= 000020          RESCVT  003152R     025 SKPSIZ  005134R   025 S.FNB = 000036          WIDTH   000072R     025
N.UNIT= 000034          RESERR= 104440          SPHOLD  000004R   024 S.FNBW= 000017          WROUND  000020R     025
OBUFLN= 001000          RESLEN  003150R     025 SPINIT  000002R   024 S.FNTY= 000004          WSDATE  000136R     025
OCHAN   005052R     025 RESPTR  003140R     025 SSTVT   000000RG  002 S.FTYP= 000002          WSFULL= 104403
OMODE   000402R     023 RESSFT  003146R     025 STKINC= 000774          S.LEN = 000002          WSNAME  000074R     025
OPTMP0  005066R     025 RESTYP  003144R     025 STOP  = 104420          S.LINK= 177774          WSTIME  000132R     025
OPTMP1  005070R     025 RESULT  003136R     025 SUBCNT  003104R   025 S.NFEN= 000020          XERROR= ****** GX
OPTMP2  005072R     025 RESVAL  003154R     025 SUBDCT  003106R   025 S.TYPE= 140000          $FIS  = ****** G
OPTMP3  005074R     025 RINC    005104R     025 SUBPTR  003110R   025 S.VAL = 000000          $FPP  = ****** G
OPTMP4  005076R     025 RLTBAS  005066R     025 SYMB1   005344R   025 TERROR  000000RG  020 $FPPLN= ****** G
OPTMP5  005100R     025 ROTINC  005070R     025 SYMF    005340R   025 TKDIMS  005072R   025 $LDFPS= ****** GX
OPTMP6  005102R     025 ROUND   000010R     025 SYMPT1  005346R   025 TKFRAM  005066R   025 $MACHN= ****** G
OPTMP7  005104R     025 ROVER   003312R     025 SYMTAB  005352R   025 TKLEFT  005070R   025 $RSTS = ****** G
OPTMP8  005106R     025 RSTS  = 000000          SYMTBF= 104404          TMODE   000403R   023 $RSX  = ****** G
OPTMP9  005110R     025 RSX   = 000115          SYMTOP  005350R   025 TMPTAG= 000166          $RT11 = ****** G
OPTM10  005112R     025 RTOP    005070R     025 SYMTP   006272R   025 TTYBUF  004140R   025 .BOOLE= 000002
OPTM14  005026R     025 RT11  = 000000          SYMWRN= 000054          TTYLEN= 000622          .CHARA= 000000
ORIGIN  000030R     025 R.BITS= 000003          SYNER   000070RG  020 TURKEY= 104436          .DEF. = 000002
OUTPTR  000066R     025 R.FIX = 000001          SYNERR= 104407          TYPE  = 000166          .FLOAT= 000006
OVROPR= 000000          R.LCT = 000002          SYSER   000056RG  020 T.BOL = 020000          .FSRCB= ****** G
PATCH   007344R     025 R.LN  = 000002          SYSERR= 104402          T.CON = 004000          .INTEG= 000004
PRECSN  000052R     025 R.LNUM= 000010          SYSTAT  000000R   024 T.DIM = 000100          .MOLUN  000404R     023
RANDAC  003064R     025 R.NEXT= 000006          SY.BAD= 000001          T.FUN = 010000          ...GBL= 000000
RANDY   003062R     025 R.OFF = 000000          SY.BRK= 100000          T.LIST= 000400          ...TPC= 000020

. ABS.  000000  000
        000000  001
TABLES  000020  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
```

```
ERROR   MACRO M1113  08-FEB-80 12:06  PAGE 47-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000074      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12939 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:15
ERROR,DM1:[305,301]ERROR/-SP=APLRX1,APLDEF,ERROR,END
```

FILEID**EXEC

```
EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC
EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC
EE           XX      XX   EE           CC
EE           XX      XX   EE           CC
EE             XX  XX     EE           CC
EE             XX  XX     EE           CC
EEEEEEEE         XX       EEEEEEEE     CC
EEEEEEEE         XX       EEEEEEEE     CC
EE             XX  XX     EE           CC
EE             XX  XX     EE           CC
EE           XX      XX   EE           CC                ....
EE           XX      XX   EE           CC                ....
EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC        ....
EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:08
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    EXEC             MAIN APL CONTROL ROUTINE
```

EXEC MACRO M1113 08-FEB-80 12:08 PAGE 1

```
                000000          RT11    =  0            ;NOT RT-11
                000000          RSTS    =  0            ;NOT RSTS/E
                000115          RSX     = 'M            ;RSX-11/M
                
                000000          EIS     =  0            ;NO EIS
                000000          FIS     =  0            ;NO FIS
                000000          FPP     =  0            ;NO FPP
                
                000004          FPPLEN  =  4.           ;SINGLE PRECISION
                
                000000          DEBUG   =  0            ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  EXEC
                                .IDENT  \V02.03\

                                .SBTTL  EXEC            MAIN APL CONTROL ROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'ESPAWN' spawns an execution frame on the
                                ;       SI stack, prompts the user, reads a line
                                ;       of input, evaluates the expression, prints
                                ;       the result, and then repeats the process.
                                ;       Any function references cause F-frames to
                                ;       be created, and if the function suspends,
                                ;       an E-frame is created ........
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     EXEC
                                ;         ..or..
                                ;       JMP     ESPAWN
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       ACOPY   - Used to copy constants
                                ;       AREAD   - Read in array (Rho 1st token on line)
                                ;       CARVE   - Return unused portion of execute buffer
                                ;       COMAND  - Used to process system commands
                                ;       EXPR    - Expression-evaluating routine
                                ;       FIXLIN  - Fixup input, if necessary
                                ;       FN      - Process function edit request
                                ;       FNBR    - A branch
                                ;       GETLIN  - Get an input line
                                ;       GETNXT  - Get next token for parsing.
                                ;       GETSAM  - Get copy of val on SI.
                                ;       KILL    - Delete result on top of the SI.
                                ;       LEX     - Interpret input line.
                                ;       PRINT   - Print result on SI (List/Value)
                                ;       PRINTM  - Print message
                                ;       SAME    - Get a copy of a value
                                ;       STKXTN  - Extend the SI stack
                                ;--

000000                          .PSECT  CODE
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 45
EXEC            MAIN APL CONTROL ROUTINE


                                        ESPAWN::
000000  005046                                    CLR     -(SP)               ;'RETURN'' ADDR FOR STKXTN
000002  023705  006346'                           CMP     SITOP,SI            ;ENOUGH SPACE LEFT?
000006  101002                                    BHI     1$                  ;NO, NOT ENOUGH
000010                                            CALL    STKXTN              ;EXTEND THE STACK
000014  162705  177772          1$:               SUB     #F.RET,SI           ;MAKE ROOM FOR RESULT
000020  013700  003314'                           MOV     SIFRAM,R0           ;POINT TO PREVIOUS FRAME
000024  001403                                    BEQ     2$                  ;IF IT EXISTS
000026  013760  004764' 000002                    MOV     FLAGS,F.FLAG(R0)    ;SAVE ITS FLAGS
000034  010015                  2$:               MOV     R0,@SI              ;SAVE PTR TO PREV FRAME
000036  010537  003314'                           MOV     SI,SIFRAM           ;POINT TO THIS FRAME
000042  012737  000100  004764'                   MOV     #F$EVAL,FLAGS       ;MAKE THIS AN E-FRAME
000050                                            POP     F.RET(SI)           ;THERE IS NO RETURN
000054  010500                                    MOV     SI,R0               ;COPY STACK POINTER
000056  010001                                    MOV     R0,R1               ;COPY POINTER
000060  062701  000020                           ADD     #F.CODE+10,R1       ;POINT TO END OF FRAME
000064  010160  000006                           MOV     R1,F.HEAD(R0)       ;SAVE POINTER IN FRAME

                                        ;+
                                        ; INITIALIZE FRAME
                                        ;-
000070  013705  003314'         EXEC::    MOV     SIFRAM,SI           ;POINT TO REF. PT.
000074  005065  177774                    CLR     F.BITS(SI)          ;CLEAR BITS
000100  005065  177776                    CLR     F.VAL(SI)           ;CLEAR VALUE
000104  052737  100000  004764'           BIS     #F.PRNT,FLAGS       ;INITIALLY PRINT LINE'S RESULTS
000112  013765  004764' 000002            MOV     FLAGS,F.FLAG(SI)    ;SAVE OUR FLAGS (TO KNOW FRAME TYPE!)
000120  005065  000004                    CLR     F.LSTC(SI)          ;CLEAR CODESTRING POINTER
000124  016505  000006                    MOV     F.HEAD(SI),SI       ;SKIP TO WHERE CODESTRING GOES

                                        ;+
                                        ; OUTPUT PROMPT CHARACTER FOR USER AND GET A LINE OF INPUT
                                        ;-
000130  013702  005052'                   MOV     OCHAN,R2            ;Point to output channel
000134  032737  000200  004764'           BIT     #F$EVI,FLAGS        ;IS THIS AN EVALUATED INPUT FRAME?
000142  001413                            BEQ     4$                  ;NOPE
000144  026262  177772  177770            CMP     C.WIDR(R2),C.WIDC(R2) ;Correct line position?
000152  001402                            BEQ     3$                  ;Yes
000154                                    CALL    NEWLIN              ;No, print <CR><LF>
000160                          3$:       QPRINT  BOXCOL              ;YES, PROMPT USER WITH BOX
000160  012701  000000'                   MOV #BOXCOL,R1              ;POINT TO BOXCOL
000170  000412                            BR      5$                  ;AND PRINT SPACES

000172  032737  000400  004764' 4$:       BIT     #F$EXEC,FLAGS       ;IS THIS AN EXECUTION FRAME?
000200  001012                            BNE     6$                  ;YES, SO NO PROMPTING
000202  026262  177772  177770            CMP     C.WIDR(R2),C.WIDC(R2) ;Correct line position?
000210  001402                            BEQ     5$                  ;Yes
000212                                    CALL    NEWLIN              ;No, print <CR><LF>
000216                          5$:       QPRINT  SPACE6              ;PRINT 6 SPACES
000216  012701  000000G                   MOV #SPACE6,R1              ;POINT TO SPACE6
000226                          6$:       CALL    GETLIN              ;GET A LINE OF INPUT
000232  032737  000400  004764'           BIT     #F$EXEC,FLAGS       ;EXECUTION FRAME?
000240  001002                            BNE     7$                  ;YES, THEN LINE IS ALREADY FIXED
000242                                    CALL    FIXLIN              ;FIX IT UP
```

```
EXEC    MACRO M1113  08-F_B-80 12:08  PAGE 46
EXEC            MAIN APL CONTROL ROUTINE


                                        ;+
                                        ; CHECK FOR DEL (FUNCTION DEF) OR RIGHT PAREN (COMMAND)
                                        ; AT START OF LINE.  TRANSFER CONTROL TO FN OR COMAND IF
                                        ; SPECIAL LINE.
                                        ;-
000246                                  7$:     PUSH    <R0,R1>         ;SAVE CHAR POINTERS
000252  112002                          8$:     MOVB    (R0)+,R2        ;GET FIRST CHAR
000254  005301                                  DEC     R1              ;DECREASE COUNT
000256  122702  000040                          CMPB    #CHRSPA,R2      ;SPACE?
000262  001773                                  BEQ     8$              ;YES, JUST IGNORE IT
000264  122702  000107                          CMPB    #CHRDEL,R2      ;IS CHAR A DEL?
000270  001403                                  BEQ     9$              ;YES, SO OK
000272  122702  000271                          CMPB    #CHRPDL,R2      ;WE ALSO ACCEPT PROTECT DELS
000276  001004                                  BNE     10$             ;NOT EITHER, YOU BOZON...
000300                                  9$:     POP     <R1,R0>         ;RESTORE INPUT POINTERS
000304  000137  000000G                        JMP     FN              ;YES, ENTER FUNCTION DEF MODE

000310  122702  000042                  10$:    CMPB    #CHRRPR,R2      ;RIGHT PAREN?
000314  001011                                  BNE     EXEC2           ;NOPE, JUST EXECUTE LINE
000316  062706  000004                          ADD     #4,SP           ;KILL SAVED INPUT POINTERS
000322                                          CALL    COMAND          ;EXECUTE SYSTEM COMMAND
000326  032737  000400  004764'                BIT     #F$EXEC,FLAGS   ;IS THIS AN EXECUTE OPERATOR?
000334  001112                                  BNE     EVAL1           ;IF SO, IT RETURNS A RESULT!!
000336  000654                                  BR      EXEC            ;AND GET A NEW LINE OF INPUT

                                        ;+
                                        ; LINE NOT SPECIAL.  EVALUATE IT
                                        ;-
000340                                  EXEC2:  POP     <R1,R0>         ;RESTORE INPUT POINTERS
000344                                          CALL    LEX             ;ANALYZE IT
000350  010205                                  MOV     R2,SI           ;RESET STACK TO TOP OF CODESTRING
000352  013700  003314'                        MOV     SIFRAME,R0      ;POINT TO THIS FRAME
000356  010560  000004                          MOV     SI,F.LSTC(R0)   ;POINT TO LAST CODESTRING BYTE
000362                                  11$:    CALL    GETNXT          ;GET FIRST TOKEN
000366                                          CALL    EXPR            ;EVALUATE THE EXPRESSION
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 47
EXEC            MAIN APL CONTROL ROUTINE


                                        ;+
                                        ; CHECK FOR CLEAN TERMINATION (BOL) OR SOMETHING SPECIAL
                                        ; SUCH AS A SEMI-COLON (SPECIAL OUTPUT FORMATTING) OR A
                                        ; BRANCH (CONTINUE FROM SUSPENDED FUNCTION).
                                        ;-
000372  032703  020000                        BIT     #T.BOL,R3       ;BEGINNING OF LINE?
000376  001071                               BNE     EVAL1           ;YES, CLEAN EXECUTION
000400  032703  040000                        BIT     #T.OTHR,R3      ;RETURN ON A CHARACTER?
000404  001465                               BEQ     16$             ;NO, SYNTAX ERROR
000406  122704  000135                        CMPB    #CHRFRO,R4      ;WAS IT A BRANCH?
000412  001034                               BNE     14$             ;NOPE
                                        ;+
                                        ; IT WAS A BRANCH...CHECK SYNTAX, KILL THIS FRAME, AND
                                        ; RESTART THE FUNCTION (FNBR).
                                        ;-
000414                                      CALL    GETNXT          ;YES, GET NEXT TOKEN
000420  032703  020000                        BIT     #T.BOL,R3       ;BOL?
000424  001455                               BEQ     16$             ;BRANCH IN EXPRESSION ILLEGAL
000426  032765  100000  177774                BIT     #T.VAL,SI.BITS(SI);EXPR HAVE A VALUE?
000434  001417                               BEQ     12$             ;NO
000436  016502  177776                        MOV     SI.VAL(SI),R2   ;GET POINTER TO DATA
000442  005762  000004                        TST     A.ALEN(R2)      ;AT LEAST 1 ITEM?
000446  001414                               BEQ     13$             ;NO
000450  013702  003314'                       MOV     SIFRAM,R2       ;POINT TO CURRENT FRAME
000454  011202                               MOV     (R2),R2         ;POINT TO PREV FRAME
000456  001406                               BEQ     12$             ;NO PREVIOUS FLAME
000460  042762  000002  000002               BIC     #F.SUSP,F.FLAG(R2);NO LONGER SUSPENDED
000466  016237  000012  005046'               MOV     F.LINE(R2),BRSELF;FLAG FOR STOPPED FN
000474  000137  000000G           12$:       JMP     FNBR            ;YES
000500  000137  000070'           13$:       JMP     EXEC            ;SPECIAL BRANCH

000504  032737  000100  004764'  14$:       BIT     #F$EVAL,FLAGS   ;EVALUATION FRAME?
000512  001422                               BEQ     16$             ;NO, THEN SEMICOLONS ARE NOT LEGAL
000514  122704  000074                        CMPB    #CHRSEM,R4      ;NO, SEMICOLON??
000520  001017                               BNE     16$             ;NO, SYNTAX ERROR
000522  032765  100000  177774                BIT     #T.VAL,SI.BITS(SI) ;CHECK FOR GARBAGE
000530  001407                               BEQ     15$             ;NO VALUE HERE
000532                                      PUSH    R2              ;SAVE POINTER
000534  014504                              MOV     -(SI),R4        ;GET THE ADDRESS OF THE DATA
000536  062705  177774                        ADD     #SI.LEN+2,SI    ;ADJUST SI TO SKIP THIS
000542                                      CALL    SAME            ;GET A NEW COPY
000546                                      POP     R2              ;RESTORE POINTER
000550  052737  100000  004764'  15$:       BIS     #F.PRNT,FLAGS   ;PRINT IT
000556  000701                               BR      11$             ;   (SPECIAL FORMATTING OF OUTPUT)

000560  104407                   16$:       SYNERR                  ;SYNTAX ERROR
```

```
                                        ;+
                                        ; LINE TERMINATED NORMALLY.  IF THIS IS AN EVALUATED INPUT
                                        ; THEN RETURN WITH THE VALUE OF THE EXPRESSION.
                                        ;-
000562                                  EVAL1::
000562  032737  002200  004764'                BIT     #F$EVI+F.END,FLAGS;THIS FRAME RETURN A VALUE?
000570  001571                                BEQ     EXEC1           ;NO, JUST PRINT RESULTS
000572  032765  100000  177774                BIT     #T.VAL,SI.BIT(SI);VAL ON SI?
000600  001004                                BNE     1$              ;YES, SO PROCESS IT
000602  032737  000200  004764'                BIT     #F$EVI,FLAGS    ;EVALUATED INPUT REQUEST?
000610  001157                                BNE     6$              ;YES, ASK AGAIN. IF NOT,
                                                                        ;   DON'T SCREW UP EXECUTE!
000612  013703  004764'                1$:     MOV     FLAGS,R3        ;SAVE CURRENT FLAGS
000616  013700  003314'                        MOV     SIFRAM,R0       ;POINT TO THIS FRAME
000622  010001                                MOV     R0,R1           ;COPY POINTER
000624  011002                                MOV     @R0,R2          ;POINT TO PREVIOUS FRAME
000626  016237  000002  004764'                MOV     F.FLAG(R2),FLAGS;GET ITS FLAGS
000634  010237  003314'                        MOV     R2,SIFRAM       ;MAKE IT THE CURRENT FRAME
000640  032765  100000  177774                BIT     #T.VAL,SI.BIT(SI); VAL ON SI?
000646  001012                                BNE     2$              ; YES, RESULT WAS RETURNED
000650                                          PUSH    <R0,R1>         ; SAVE REGS
000654                                          CALL    KILL            ; WIPE OUT EMPTY RESULT FRAME
000660  012704  007364'                        MOV     #NULL,R4        ; POINT TO NULL ARRAY
000664                                          CALL    SAME            ; COPY IT TO MAKE IT THE RESULT
000670                                          POP     <R1,R0>         ; RESTORE REGS AND RETURN NULL RESULT
                                        ;+
                                        ; IF RESULT IS NOT A TEMPORARY, IE. IT IS A VARIABLE,
                                        ; THE RESULT MUST BE COPIED SO THAT WHOEVER CALLED US DOESN'T
                                        ; CLOBBER THE VARIABLE.
                                        ;-
000674                                  2$:     PUSH    <R0,R1,R3>      ;SAVE POINTERS
000702  016504  177776                        MOV     SI.VAL(SI),R4   ;POINT TO VALUE
000706                                          CALL    SAME            ;COPY IT IF NECESSARY
000712  014504                                MOV     -(SI),R4        ;NEW VALUE
000714  014503                                MOV     -(SI),R3        ;NEW BITS
000716  005745                                TST     -(SI)           ;RETURN
000720  020465  177776                        CMP     R4,SI.VAL(SI)   ;BEING DONE IN PLACE (TEMPORARY RESULT?)
000724  001406                                BEQ     3$              ;YES, SO NO BOTHER
000726                                          CALL    KILL            ;ELSE KILL OLD FRAME
000732  012725  000000G                        MOV     #SYSER,(SI)+    ;FAKE RETURN
000736  010325                                MOV     R3,(SI)+        ;NEW BITS
000740  010425                                MOV     R4,(SI)+        ;NEW VALUE
000742                                  3$:     POP     <R3,R1,R0>      ;RESTORE REGS
000750  014540                                MOV     -(SI),-(R0)     ;FILL IN RESULT
000752  014540                                MOV     -(SI),-(R0)     ;FILL IN RESULT BITS
000754  016100  000006                        MOV     F.HEAD(R1),R0   ;POINT TO END OF HEADER
000760  014037  005056'                        MOV     -(R0),E$CT      ;RESTORE INPUT POINTERS
000764  014037  005060'                        MOV     -(R0),E$PT      ;TO ALLOW NESTING
000770  014037  000064'                        MOV     -(R0),INPTR     ;OF EXECUTION OPERATORS
000774  014046                                MOV     -(R0),-(SP)     ;RESTORE RETURN
000776  014046                                MOV     -(R0),-(SP)     ;RESTORE R4
001000  014046                                MOV     -(R0),-(SP)     ;RESTORE R3
001002  014046                                MOV     -(R0),-(SP)     ;RESTORE R2
001004  010105                                MOV     R1,SI           ;KILL THE EVAL/EVI FRAME
                                        ;+
                                        ; CHECK FOR EXECUTION FRAME.  (YECH!)
                                        ;-
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 48-1
EXEC            MAIN APL CONTROL ROUTINE

    250 001006  032703  000400                     BIT     #F$EXEC,R3      ;WAS THIS AN EXECUTION FRAME?
    251 001012  001424                             BEQ     4$              ;NOPE
    252 001014  013700  003040'                    MOV     CHANEX+C.BUFF,R0;GET RESULT BUFFER
    253 001020  001421                             BEQ     4$              ;DOES NOT EXIST
    254 001022  010002                             MOV     R0,R2           ;COPY POINTERS
    255 001024  013701  003044'                    MOV     CHANEX+C.CPTR,R1;LAST ENTRY
    256 001030  161001                             SUB     (R0),R1         ;LENGTH OF DATA
    257 001032  010162  000004                     MOV     R1,A.ALEN(R2)   ;FIX LENGTH
    258 001036  010162  000006                     MOV     R1,A.DIM1(R2)   ;OF RESULT CHARACTER VECTOR
    259 001042                                     CALL    CARVE           ;RETURN UNUSED PORTION
    260 001046  012737  000256' 005052'            MOV     #CHAN00,OCHAN   ;OUTPUT TO TTY NOW
    261 001054  005037  003040'                    CLR     CHANEX+C.BUFF   ;PREVENT PROBLEMS OF DEALLOCATION
    262 001060  005037  003060'                    CLR     CHANEX          ;KILL THE CHANNEL
    263                                    ;+
    264                                    ; CHECK IF VALUE IS A CONSTANT.  IF SO, THE VALUE MUST BE
    265                                    ; COPIED BECAUSE THE CONSTANT WILL GO AWAY
    266                                    ; (IN FACT IT ALREADY HAS!)
    267                                    ;-
    268 001064  032765  100000  177774  4$:    BIT     #T.VAL,SI.BIT(SI);VAL ON SI?
    269 001072  001422                             BEQ     5$              ;NOPE, NOTHING TO DO
    270 001074  016500  177776                     MOV     SI.VAL(SI),R0   ;GET VALUE
    271 001100  132760  000020  000003             BITB    #A.CON,A.BITS(R0);CONSTANT?
    272 001106  001414                             BEQ     5$              ;NOPE, GOOD!
    273 001110                                     PUSH    R0              ;SAVE PTR TO CONSTANT
    274 001112                                     CALL    GETSAM          ;GET IDENTICAL ARRAY
    275 001116  062705  177772                     ADD     #SI.LEN,SI      ;KILL PARSE FRAME CREATED BY 'GETSAM'!!
    276 001122  010065  177776                     MOV     R0,SI.VAL(SI)   ;SET NEW VALUE
    277 001126                                     POP     R2              ;GET BACK PTR TO CONSTANT
    278 001130  016200  000004                     MOV     A.ALEN(R2),R0   ;GET LENGTH TO COPY
    279 001134                                     CALL    ACOPY           ;COPY THE ARRAY
    280
    281 001140  005037  003112'            5$:     CLR     DIMBLK          ;NEVER DIM ARG FOR EXECUTE OR QUAD!
    282 001144  000175  177772                     JMP     @SI.RET(SI)     ;AND RETURN TO CALLER
    283
    284 001150  000137  000070'            6$:     JMP     EXEC            ;INDIRECT JUMP CAUSE TOO FAR
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 49
EXEC            MAIN APL CONTROL ROUTINE


                                        ;+
                                        ; THIS IS JUST AN EVALUATION FRAME.  PRINT ALL OF THE
                                        ; EXPRESSIONS ON THE STACK (IF PRINT FLAG IS SET), KILL
                                        ; THE PARSE FRAMES, AND START ALL OVER.
                                        ;-
001154  013703  003314'         EXEC1:  MOV     SIFRAM,R3       ;POINT TO THIS FRAME
001160  016303  000004                  MOV     F.LSTC(R3),R3   ;POINT TO START OF PARSE FRAMES
001164  001004                          BNE     1$              ;THERE IS A CODESTRING
001166  013703  003314'                  MOV     SIFRAM,R3       ;ELSE WE MUST USE HEADER POINTER
001172  016303  000006                  MOV     F.HEAD(R3),R3   ;TO POINT TO END OF PARSE FRAMES
001176  020503                  1$:     CMP     SI,R3           ;ANY PARSE FRAMES ON SI?
001200  003012                          BGT     4$              ;YES, SO PRINT/KILL IT
001202  001010                          BNE     3$              ;OOPS!
001204  032737  100000  004764'         BIT     #F.PRNT,FLAGS   ;ALREADY AT HEAD OF LINE?
001212  001402                          BEQ     2$              ;YES, DON'T NEED ANOTHER LINE FEED
001214                                  CALL    NEWLIN          ;Print a <CR><LF> pair
001220  000137  000070'          2$:     JMP     EXEC            ;ALL DONE

001224  104402                  3$:     SYSERR                  ;HORRIBLE SI STACK SCREW UP!

001226  032737  100000  004764' 4$:     BIT     #F.PRNT,FLAGS   ;PRINT RESULT?
001234  001403                          BEQ     5$              ;NO, JUST KILL IT
001236  010500                          MOV     SI,R0           ;POINT TO RESULT
001240                                  CALL    PRINT           ;PRINT RESULT OF EXPRESSION
001244                          5$:     CALL    KILL            ;KILL RESULT OF EXPRESSION
001250  000752                          BR      1$              ;AND PRINT EVERYTHING LEFT

000000                                  .PSECT  TABLES

000000                          BOXCOL::
000000  114  076  015                   .ASCIZ  <CHRBOX><CHRCOL><CHR.CR><CHR.LF>
000003  012  000
                                        .EVEN
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 51
EXEC            MAIN APL CONTROL ROUTINE

     1          000001                          .END
```

EXEC MACRO M1113 08-FEB-80 12:08 PAGE 51-1
SYMBOL TABLE

```
ACOPY = ****** GX        CHRERR= 104431      CTMAX   000212R  025 D.CONT= 000006      EDICNT  005064R  025
AC0   =%000000           CHRFIL= 000177      CTRMAX  000172R  025 D.CTLM= 000000      EDIFLG  005062R  025
AC1   =%000001           CHRFRO= 000135          C.ALOC= 000162      D.DEAL= 000003      EDIT  = 000003
AC2   =%000002           CHRLBR= 000073          C.BLOK= 000064      D.DECD= 000002      EIS   = 000000
AC3   =%000003           CHRLPR= 000072          C.BLSZ= 000140      D.DIV = 000000      ELESIZ  005132R  025
AC4   =%000004           CHRMNS= 000137          C.BUFF= 177760      D.DMD = 000010      ENDCOR  000004R  025
AC5   =%000005           CHRMUL= 000075          C.BUFL= 177762      D.DQAD= 177777      EOF   = 104475
ALLINT  003274R  025     CHRNEG= 000100          C.CCNT= 177766      D.DROP= 000002      ERRMAX= 000166
ARYRET  003276R  025     CHRPDL= 000271          C.CPTR= 177764      D.ENCD= 000002      ERRNUM  004772R  025
ASCQUO= 000042           CHRPLU= 000055          C.CZ  = 001000      D.EQ  = 000000      ERRPTR  004770R  025
A.ALEN= 000004           CHRPWR= 000120          C.DIG = 000040      D.EXPD= 000002      ERR0    004774R  025
A.ARG = 000040           CHRQST= 000121          C.FASW= 020000      D.FGET= 000005      ESPAWN  000000RG  020
A.ASB = 000200           CHRQUO= 000113          C.FLEN= 000010      D.FMT = 000004      EVAL1   000562RG  020
A.BITS= 000003           CHRRBR= 000047          C.FLSH= 004000      D.FPNT= 000005      EXEBUF  005054R  025
A.CON = 000020           CHRRPR= 000042          C.FRC = 000001      D.FPUT= 000005      EXEC    000070RG  020
A.DIMS= 000002           CHRSEM= 000074          C.IDEN= 000010      D.GE  = 000000      EXEC1   001154R  020
A.DIM1= 000006           CHRSMO= 000112          C.NAME= 000102      D.GT  = 000000      EXEC2   000340R  020
A.NUM = 000006           CHRSPA= 000040          C.NOBF= 000400      D.IBM = 000002      EXPR  = ****** GX
A.PTR = 000000           CHRUPA= 000131          C.NUM = 000020      D.IND = 000002      E$CT    005056R  025
A.TMP = 000010           CHR.BS= 000010          C.OP  = 000004      D.INNR= 000013      E$PT    005060R  025
BADDIR= 104463           CHR.CC= 000003          C.OPN = 010000      D.INTR= 000006      FA.APD= 000100
BADERR= 104437           CHR.CR= 000015          C.RDLN= 177776      D.LE  = 000000      FA.CRE= 000010
BLZBUB  005116R  025     CHR.CZ= 000032          C.REWT= 002000      D.LOG = 000001      FA.DLK= 001000
BOXCOL  000000RG 002     CHR.GT= 000046          C.RND = 000010      D.LT  = 000000      FA.ENB= 100000
BRSELF  005046R  025     CHR.LF= 000012          C.SBIT= 177756      D.MAX = 000000      FA.EXC= 002000
CARVE = ****** GX        CHR.LT= 000043          C.SPFX= 000100      D.MEM = 000002      FA.EXT= 000004
CCNT    005072R  025     CHR.SI= 000017          C.SPGL= 000200      D.MIN = 000000      FA.NSP= 000100
CHANEX  003060R  025     CHR.SL= 000057          C.SPLX= 000001      D.MUL = 000000      FA.NSR= 000200
CHAN00  000256R  025     CHR.SO= 000016          C.SPRS= 000002      D.NE  = 000000      FA.POS= 010000
CHAN01  000330R  025     CHR.TB= 000011          C.SRC = 177774      D.NN  = 000000      FA.RD = 000001
CHAN02  000512R  025     CHR.UL= 000106          C.SRCP= 000024      D.NR  = 000000      FA.RWD= 004000
CHAN03  000674R  025     CHR..A= 000141          C.TT  = 000004      D.OR  = 000000      FA.SEQ= 040000
CHAN04  001056R  025     CHR..E= 000145          C.WIDC= 177770      D.PLUS= 000000      FA.SHR= 000040
CHAN05  001240R  025     CHR..S= 000163          C.WIDR= 177772      D.PSUB= 000006      FA.TMP= 000020
CHAN06  001422R  025     CHR..T= 000164          DEBUG = 000000      D.PWR = 000001      FA.WCK= 020000
CHAN07  001604R  025     CHR..U= 000165          DEFERR= 104405      D.QQAD= 000000      FA.WRT= 000002
CHAN08  001766R  025     CHR..Z= 000172          DEVIL   005114R  025 D.QUAD= 000000      FD.BLK= 000010
CHAN09  002150R  025     CHR..0= 000060          DEVNTF= 104470      D.RES = 000000      FD.CCL= 000002
CHAN10  002332R  025     CHR..1= 000061          DIGITS  000006R  025 D.RHO = 000000      FD.COM= 020000
CHAN11  002514R  025     CH.AND= 000001          DIMAX = 000020      D.ROT = 000003      FD.CR = 000002
CHAN12  002676R  025     CMDBUF  003250R  025    DIMBLK  003112R  025 D.SCON= 000006      FD.DIR= 000010
CHAN13  000022R  023     CMDMAX= 000010          DIMPTR  003114R  025 D.SDIF= 000006      FD.FTN= 000001
CHAN14  000204R  023     CNTSIZ  005130R  025    DIMTYP  003116R  025 D.SUB = 000000      FD.F11= 040000
CHNOPN= 104471           COMAND= ****** GX       DIRFUL= 104466      D.SUBS= 000006      FD.INS= 000010
CHRAND= 000051           COMBIN  005076R  025    DOMERR= 104417      D.TAKE= 000002      FD.ISP= 002000
CHRBEL= 000007           COMCNT  005126R  025    DPTERR= 104416      D.TRAN= 000004      FD.MNT= 100000
CHRBKR= 000133           COMERR= 104425          DYDCNT  003224R  025 D.UNON= 000006      FD.OSP= 004000
CHRBOL= 000200           COMPTR  005122R  025    D.AND = 000000      D.$MAX= 000000      FD.PLC= 000004
CHRBOX= 000114           COMRTN  003156R  025    D.BXCT= 000007      D.$MIN= 000000      FD.PRN= 000004
CHRCMA= 000054           COMSKP  005124R  025    D.BXIO= 000007      D.$MUL= 000000      FD.PSE= 010000
CHRCOL= 000076           CORPTR  005120R  025    D.BXNL= 000007      D.$PLU= 000000      FD.RAH= 000001
CHRDEL= 000107           CS.ASS= 000200          D.BXPP= 000007      D.$RES= 000000      FD.RAN= 000002
CHRDIV= 000053           CS.BOO= 000010          D.BXPW= 000007      D.$SUB= 000000      FD.REC= 000001
CHRDLR= 000176           CS.FLO= 000020          D.BXRL= 000007      D.1COM= 000001      FD.RWM= 000001
CHRDLT= 000110           CS.IDN= 000040          D.CIRC= 000001      D.1EXP= 000002      FD.SDI= 000020
CHRDOT= 000056           CS.ONE= 000100          D.COMB= 000001      D.1ROT= 000003      FD.SQD= 000040
CHREP = 000105           CTEPS   000152R  025    D.COMP= 000001      EDICHR  005063R  025 FD.TTY= 000004
```

EXEC MACRO M1113 08-FEB-80 12:08 PAGE 51-2
SYMBOL TABLE

```
FD.WBH= 000002          F.DSPT= 000044          F.TRC = 000002          LNLNG = 104541          M.MMD = 000010
FF.CHR= 000005          F.DVNM= 000134          F.UNIT= 000136          LOCLCT  005036R 025     M.NEG = 000000
FF.NV = 000003          F.EFBK= 000010          F.URBD= 000020          LOWCOR  000002R 025     M.NOT = 000000
FF.POE= 000002          F.EFN = 000050          F.VAL = 177776          LSTERP  004766R 025     M.NUB = 000006
FF.RWD= 000001          F.END = 002000          F.VBN = 000064          LWIDTH  000070R 025     M.NUM = 000204
FF.RWF= 000006          F.EOBB= 000032          F.VBSZ= 000060          L.BITS= 000003          M.OUTR= 000012
FF.SPC= 000004          F.ERR = 000052          GETLIN= ****** GX       L.CNT = 000002          M.PITM= 000000
FHEADR  005012R 025     F.FACC= 000043          GETNXT= ****** GX       L.PRNT= 000001          M.PRDS= 000006
FHRET   005040R 025     F.FFBY= 000014          GETSAM= ****** GX       L.PTR = 000000          M.QQAD= 000000
FILNTF= 104467          F.FLAG= 000002          GNOME   003102R 025     L.PTR1= 000004          M.QUAD= 000000
FIS   = 000000          F.FLB = 000100          HASHC   005136R 025     L.SUBS= 000002          M.RAND= 000012
FISCON= 046000          F.FNAM= 000110          HASHTB  005140R 025     MACHIN= 000106          M.RAVL= 000000
FIXLIN= ****** GX       F.FNB = 000102          HSHMSK= 177601          MAXCHN= 000014          M.RCIP= 000000
FIXPTR  003074R 025     F.FNDF= 000010          HSHSIZ= 000200          MONBLK  003120R 025     M.RDCT= 000011
FLAGS   004764R 025     F.FTYP= 000116          ICHAN   005050R 025     MONCVT  003132R 025     M.REVS= 000003
FLCBUF  003250R 025     F.FUN = 000010          ILFNAM= 104464          MONLEN  003130R 025     M.ROLL= 000000
FLINE   005014R 025     F.FVER= 000120          ILLCHN= 104540          MONPTR  003122R 025     M.SATA= 000001
FLOEXP  003270R 025     F.HEAD= 000006          ILLSW = 104565          MONSFT  003126R 025     M.SCAN= 000011
FLONEG  003272R 025     F.HIBK= 000004          INDERR= 104410          MONTYP  003124R 025     M.SHAP= 000000
FLSWTB  000222R 025     F.LBL = 000020          INNTMP  003250R 025     MONVAL  003134R 025     M.SIGN= 000000
FLTLEN= 000010          F.LINE= 000012          INNTM2  005016R 025     M.ABS = 000000          M.SUBR= 000001
FN    = ****** GX       F.LOC = 000004          INPTR   000064R 025     M.BOL = 000200          M.SYM = 000203
FNBR  = ****** GX       F.LSTC= 000004          INSNAM= 104462          M.BXAV= 000007          M.SYN = 000201
FNINSI  005044R 025     F.LUN = 000042          IOERR = 104477          M.BXCR= 000007          M.TRAN= 000012
FORIGN  000032R 025     F.MBCT= 000054          I.CNT = 177776          M.BXCT= 000007          M.$ABS= 000000
FO.APD= 000106          F.MBC1= 000055          I.DIM = 177772          M.BXEX= 000007          M.$CEI= 000000
FO.MFY= 000002          F.MBFG= 000056          I.LEN = 000012          M.BXFX= 000007          M.$FLR= 000000
FO.RD = 000001          F.MON = 000002          I.OERR= 000062          M.BXIO= 000007          M.$IDN= 000000
FO.UPD= 000006          F.NIL = 000001          I.OLD = 177770          M.BXLC= 000003          M.$NEG= 000000
FO.WRT= 000016          F.NRBD= 000024          I.PTR = 177774          M.BXNC= 000007          M.$SIG= 000000
FPMODE  000042R 025     F.NREC= 000030          I.STR = 177766          M.BXNL= 000007          M.1RDT= 000011
FPP   = 000000          F.OPN = 000010          KILL  = ****** GX       M.BXPP= 000007          M.1REV= 000003
FPPLEN= 000004          F.OVBS= 000030          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FREBLK  007400R 025     F.PREV= 000000          LABLPT  005042R 025     M.BXRL= 000007          NAMBUF  003240R 025
FUZZ    000142R 025     F.PRNT= 100000          LARGBL  003162R 025     M.BXTS= 000007          NB.DEV= 000200
FXPTR1  003076R 025     F.PROT= 010000          LARGCV  003176R 025     M.BXWA= 000007          NB.DIR= 000100
F$EVAL= 000100          F.RACC= 000016          LARGFL  003160R 025     M.CEI = 177777          NB.NAM= 000004
F$EVI = 000200          F.RATT= 000001          LARGLE  003174R 025     M.CHR = 000202          NB.SD1= 000400
F$EXEC= 000400          F.RCNM= 000034          LARGLN  003166R 025     M.CMT = 000206          NB.SD2= 001000
F$FUNC= 001000          F.RCTL= 000017          LARGPT  003164R 025     M.DOUT= 000006          NB.SNM= 000040
F.ACTL= 000076          F.READ= 004000          LARGSF  003172R 025     M.DQAD= 000006          NB.STP= 000020
F.ALOC= 000040          F.REDF= 000040          LARGTY  003170R 025     M.ENCD= 000004          NB.SVR= 000010
F.BBFS= 000062          F.RES = 000040          LARGVA  003200R 025     M.EXE = 000000          NB.TYP= 000002
F.BDB = 000070          F.RET = 177772          LBINC   005100R 025     M.EXP = 000001          NB.VER= 000001
F.BGBC= 000057          F.RSIZ= 000002          LBUFF   003316R 025     M.FACT= 000001          ND.DEV= 000200
F.BITS= 177774          F.RTYP= 000000          LBUFLN= 000621          M.FIL = 000177          ND.EXT= 000002
F.BKDN= 000026          F.R2  = 000014          LCOUNT  005106R 025     M.FLR = 177777          ND.NAM= 000004
F.BKDS= 000020          F.R3  = 000016          LENERR= 104412          M.FMT = 000004          ND.PPN= 000100
F.BKEF= 000050          F.R4  = 000020          LENTAB  000054R 025     M.GDUP= 000003          ND.$EX= 000000
F.BKP1= 000051          F.SEQN= 000100          LEX   = ****** GX       M.GDWN= 000003          ND.$ES= 000020
F.BKST= 000024          F.SPDV= 000072          LEXPTR  003100R 025     M.GETI= 000006          ND.$NM= 000000
F.BKVB= 000064          F.SPUN= 000074          LFTOP   005066R 025     M.IBM = 000003          ND.$NS= 000040
F.BR  = 000020          F.STBK= 000036          LINC    005102R 025     M.IDN = 000000          ND.$PN= 001400
F.CHR = 000075          F.STK = 000022          LINE    005016R 025     M.INSP= 000006          NEGSGN  005063R 025
F.CNTG= 000034          F.STOP= 000004          LININC  005026R 025     M.IOTA= 000000          NEWLIN= ****** GX
F.CODE= 000010          F.STP = 000001          LISIZE= 000020          M.LIT = 000205          NINPT1  003302R 025
F.DFNB= 000046          F.SUSP= 000002          LITOP   005072R 025     M.LN  = 000001          NINPT2  003300R 025
```

EXEC MACRO M1113 08-FEB-80 12:08 PAGE 51-3
SYMBOL TABLE

```
NINSTK  003304R     025 OPTMP7  005104R     025 RSTS   = 000000          SYMTOP  005350R     025 T.BOL = 020000
NINTMP  003310R     025 OPTMP8  005106R     025 RSX    = 000115          SYMTP   006272R     025 T.CON = 004000
NINTYP  003306R     025 OPTMP9  005110R     025 RTOP     005070R     025 SYMWRN= 000054          T.DIM = 000100
NOTAVL= 104472          OPTM10  005112R     025 RT11   = 000000          SYNERR= 104407          T.FUN = 010000
NOTOPN= 104473          OPTM14  005026R     025 R.BITS= 000003           SYSER = ****** GX       T.LIST= 000400
NPRT1   004762R     025 ORIGIN  000030R     025 R.FIX  = 000001          SYSERR= 104402          T.LIT = 000020
NULL    007364R     025 OUTPTR  000066R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.MON = 000010
NUMCHN= 000020          OVROPR= 000000          R.LN   = 000002          SY.BAD= 000001          T.NIL = 000004
NUMDFL  003230R     025 PATCH   007344R     025 R.LNUM= 000010           SY.BRK= 100000          T.NUM = 002000
NUMEFL  003233R     025 PRECSN  000052R     025 R.NEXT= 000006           SY.BS = 000010          T.OPFN= 000001
NUMESN  003236R     025 PRINT = ****** GX       R.OFF  = 000000          SY.CLR= 000040          T.OTHR= 040000
NUMFLG  003230R     025 PRINTM= ****** GX       R.PREV= 000004           SY.ERP= 000020          T.SOP = 000002
NUMLEN  000062R     025 RANDAC  003064R     025 R.PTR  = 000000          SY.NOU= 000100          T.SPOP= 000040
NUMNEG  003232R     025 RANDY   003062R     025 R.SEQ  = 000003          SY.OPN= 000002          T.TMP = 000200
NUMQFL  003234R     025 RANERR= 104411          R.VAR  = 000002          SY.PMT= 001000          T.VAL = 100000
NUMRIG  003231R     025 RARGBL  003204R     025 R6     =%000006          SY.SRT= 000400          T.VAR = 001000
NUMWID  003226R     025 RARGCV  003220R     025 R7     =%000007          SY.SVC= 000004          UPPER = 177400
NUMZFL  003235R     025 RARGFL  003202R     025 SAME   = ****** GX       SY.SVL= 000200          UTNAME  000344R     023
NYIERR= 104432          RARGLE  003216R     025 SI     =%000005          S.AB   = 177776          VALERR= 104413
N.BLSZ= 000036          RARGLN  003210R     025 SIDAMN= 104433           S.ASB  = 100000          VARBLE  007400R     025
N.DESC= 000020          RARGPT  003206R     025 SIFRAM  003314R     025 S.BFHD= 000020          VARBLS  000000R     025
N.DEV = 000032          RARGSF  003214R     025 SISTAK  006350R     025 S.BITS= 000002          VERNUM  000000R     025
N.DEVN= 000034          RARGTY  003212R     025 SITOP   006346R     025 S.CHR1= 000003          VRSION= 000002
N.DID = 000024          RARGVA  003222R     025 SITP    007144R     025 S.FATT= 000016          WIDTH   000072R     025
N.DVNM= 000032          RBASE   005074R     025 SIWARN= 000200           S.FDB  = 000140          WROUND  000020R     025
N.EXT = 000014          RCOUNR  005112R     025 SIZTAB  000044R     025 S.FLB  = 040000          WSDATE  000136R     025
N.FID = 000000          RCOUNT  005110R     025 SI.BIT= 177774           S.FNAM= 000006          WSFULL= 104403
N.FNAM= 000006          REDCNT  005070R     025 SI.LEN= 177772           S.FNB  = 000036          WSNAME  000074R     025
N.FTYP= 000014          REDIGN  005066R     025 SI.RET= 177772           S.FNBW= 000017          WSTIME  000132R     025
N.FVER= 000016          REDOFF  005072R     025 SI.VAL= 177776           S.FNTY= 000004          $FIS   = ****** G
N.NAME= 000006          REENTR  003060R     025 SKPSIZ  005134R     025 S.FTYP= 000002          $FPP   = ****** G
N.NEXT= 000022          RESBIT  003142R     025 SPACE6= ****** GX        S.LEN  = 000002          $FPPLN= ****** G
N.PPN = 000024          RESCVT  003152R     025 SPHOLD  000004R     024 S.LINK= 177774          $MACHN= ****** G
N.STAT= 000020          RESEDR= 104440          SPINIT  000002R     024 S.NFEN= 000020          $RSTS  = ****** G
N.UNIT= 000034          RESILN  003150R     025 STKINC= 000774           S.TYPE= 140000          $RSX   = ****** G
OBUFLN= 001000          RESVR   003140R     025 STKXTN= ****** GX        S.VAL  = 000000          $RT11  = ****** G
OCHAN   005052R     025 RESSFT  003146R     025 STOP   = 104420          TKDIMS  005072R     025 .BOOLE= 000002
OMODE   000402R     023 RESTYP  003144R     025 SUBCNT  003104R     025 TKFRAM  005066R     025 .CHARA= 000000
OPTMP0  005066R     025 RESULT  003136R     025 SUBDCT  003106R     025 TKLEFT  005070R     025 .DEF. = 000002
OPTMP1  005070R     025 RESVAL  003154R     025 SUBPTR  003110R     025 TMODE   000403R     023 .FLOAT= 000006
OPTMP2  005072R     025 RINC    005104R     025 SYMB1   005344R     025 TMPTAG= 000166          .FSRCB= ****** G
OPTMP3  005074R     025 RLTBAS  005066R     025 SYMF    005340R     025 TTYBUF  004140R     025 .INTEG= 000004
OPTMP4  005076      025 ROTINC  005070R     025 SYMPT1  005346R     025 TTYLEN= 000622          .MOLUN  000404R     023
OPTMP5  005100      025 ROUND   000010R     025 SYMTAB  005352R     025 TURKEY= 104436          ...GBL= 000000
OPTMP6  00510       025 ROVER   003312R     025 SYMTBF= 104404          TYPE   = 000166          ...TPC= 000020

. ABS.  000000  000
        000000  001
TABLES  000006  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
```

```
EXEC    MACRO M1113  08-FEB-80 12:08  PAGE 51-4
SYMBOL TABLE

ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    001252      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13146 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:42
EXEC,DM1:[305,301]EXEC/-SP=APLRX1,APLDEF,EXEC,END
```

FILEID**EXISTS

```
EEEEEEEEEE   XX      XX     IIIIII      SSSSSSSS   TTTTTTTTTT     SSSSSSSS
EEEEEEEEEE   XX      XX     IIIIII      SSSSSSSS   TTTTTTTTTT     SSSSSSSS
EE           XX      XX       II      SS             TT       SS
EE           XX      XX       II      SS             TT       SS
EE             XX  XX         II      SS             TT       SS
EE             XX  XX         II      SS             TT       SS
EEEEEEEE         XX           II        SSSSSS       TT         SSSSSS
EEEEEEEE         XX           II        SSSSSS       TT         SSSSSS
EE             XX  XX         II              SS     TT               SS
EE             XX  XX         II              SS     TT               SS
EE           XX      XX       II              SS     TT               SS      ....
EE           XX      XX       II              SS     TT               SS      ....
EEEEEEEEEE   XX      XX     IIIIII      SSSSSSSS     TT       SSSSSSSS      ....
EEEEEEEEEE   XX      XX     IIIIII      SSSSSSSS     TT       SSSSSSSS      ....

LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:11
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    EXISTS                  VERIFY IF A FILE EXISTS
```

```
EXISTS  MACRO M1113  08-FEB-80 12:11  PAGE 1


      1        000000                  RT11    =  0          ;NOT RT-11
      2        000000                  RSTS    =  0          ;NOT RSTS/E
      3        000115                  RSX     = 'M          ;RSX-11/M
      4
      5        000000                  EIS     =  0          ;NO EIS
      6        000000                  FIS     =  0          ;NO FIS
      7        000000                  FPP     =  0          ;NO FPP
      8
      9        000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 12:11  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
EXISTS  MACRO M1113  08-FEB-80 12:11  PAGE 44
DEFINITIONS     ERRORS

                                .TITLE  EXISTS
                                .IDENT  \V02.03\

                                .SBTTL  EXISTS                  VERIFY IF A FILE EXISTS

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       THIS ROUTINE VERIFIES THAT A FILE WHOSE NAME BLOCK
                                ;       IS POINTED TO BY R4 DOES IN FACT EXIST
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    EXISTS
                                ;
                                ; INPUTS:
                                ;       R4      - POINTS TO THE NAME BLOCK OF THE FILE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       Condition Codes - C-Bit Cleared if File Found, Set Otherwise
                                ;
                                ; SIDE EFFECTS:
                                ;       For RSTS, the File is Opened on Channel 14.
                                ;--

000000                                  .PSECT  CODE

000000                          EXISTS::
000000                                  PUSH    R2              ;Save a register
000002  012702  000204'                 MOV     #CHAN14,R2      ;USE UTILITY CHANNEL
000006                                  CALL    RESCHN          ;CLEAN IT OUT (JUST IN CASE)
                                ;
                                ; RSX Check, see if File exists
                                ;
000012  010401                          MOV     R4,R1           ;Point to Name Block
000014  010200                          MOV     R2,R0           ;Point to FDB
000016                                  POP     R2              ;Restore register
000020                                  CALLR   .FIND           ;Find it, Please
```

```
EXISTS  MACRO M1113  08-FEB-80 12:11  PAGE 46
EXISTS                    VERIFY IF A FILE EXISTS

      1         000001                              .END
```

```
EXISTS  MACRO M1113  08-FEB-80 12:11  PAGE 46-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073  C.BLSZ= 000140  D.DIV = 000000  ELESIZ  005132R  025
AC1    =%000001          CHRLPR= 000072  C.BUFF= 177760  D.DMD = 000010  ENDCOR  000004R  025
AC2    =%000002          CHRMNS= 000137  C.BUFL= 177762  D.DQAD= 177777  EOF    = 104475
AC3    =%000003          CHRMUL= 000075  C.CCNT= 177766  D.DROP= 000002  ERRMAX= 000166
AC4    =%000004          CHRNEG= 000100  C.CPTR= 177764  D.ENCD= 000002  ERRNUM  004772R  025
AC5    =%000005          CHRPDL= 000271  C.CZ   = 001000  D.EQ  = 000000  ERRPTR  004770R  025
ALLINT  003274R  025    CHRPLU= 000055  C.DIG  = 000040  D.EXPD= 000002  ERR0    004774R  025
ARYRET  003276R  025    CHRPWR= 000120  C.FASW= 020000  D.FGET= 000005  EXEBUF  005054R  025
ASCQUO= 000042          CHRQST= 000121  C.FLEN= 000010  D.FMT = 000004  EXISTS  000000RG  020
A.ALEN= 000004          CHRQUO= 000113  C.FLSH= 004000  D.FPNT= 000005  E$CT    005056R  025
A.ARG = 000040          CHRRBR= 000047  C.FRC  = 000001  D.FPUT= 000005  E$PT    005060R  025
A.ASB = 000200          CHRRPR= 000042  C.IDEN= 000010  D.GE  = 000000  FA.APD= 000100
A.BITS= 000003          CHRSEM= 000074  C.NAME= 000102  D.GT  = 000000  FA.CRE= 000010
A.CON = 000020          CHRSMO= 000112  C.NOBF= 000400  D.IBM = 000002  FA.DLK= 001000
A.DIMS= 000002          CHRSPA= 000040  C.NUM  = 000020  D.IND = 000002  FA.ENB= 100000
A.DIM1= 000006          CHRUPA= 000131  C.OP   = 000004  D.INNR= 000013  FA.EXC= 002000
A.NUM = 000006          CHR.BS= 000010  C.OPN  = 010000  D.INTR= 000006  FA.EXT= 000004
A.PTR = 000000          CHR.CC= 000003  C.RDLN= 177776  D.LE  = 000000  FA.NSP= 000100
A.TMP = 000010          CHR.CR= 000015  C.REWT= 002000  D.LOG = 000001  FA.NSR= 000200
BADDIR= 104463          CHR.CZ= 000032  C.RND  = 000010  D.LT  = 000000  FA.POS= 01000(
BADERR= 104437          CHR.GT= 000046  C.SBIT= 177756  D.MAX = 000000  FA.RD  = 000001
BLZBUB  005116R  025    CHR.LF= 000012  C.SPFX= 000100  D.MEM = 000002  FA.RWD= 004000
BRSELF  005046R  025    CHR.LT= 000043  C.SPGL= 000200  D.MIN = 000000  FA.SEQ= 040000
CCNT    005072R  025    CHR.SI= 000017  C.SPLX= 000001  D.MUL = 000000  FA.SHR= 000040
CHANEX  003060R  025    CHR.SL= 000057  C.SPRS= 000002  D.NE  = 000000  FA.TMP= 000020
CHAN00  000256R  025    CHR.SO= 000016  C.SRC  = 177774  D.NN  = 000000  FA.WCK= 020000
CHAN01  000330R  025    CHR.TB= 000011  C.SRCP= 000024  D.NR  = 000000  FA.WRT= 000002
CHAN02  000512R  025    CHR.UL= 000106  C.TT   = 000004  D.OR  = 000000  FD.BLK= 000010
CHAN03  000674R  025    CHR..A= 000141  C.WIDC= 177770  D.PLUS= 000000  FD.CCL= 000002
CHAN04  001056R  025    CHR..E= 000145  C.WIDR= 177772  D.PSUB= 000006  FD.COM= 020000
CHAN05  001240R  025    CHR..S= 000163  DEBUG  = 000000  D.PWR = 000001  FD.CR  = 000002
CHAN06  001422R  025    CHR..T= 000164  DEFERR= 104405  D.QQAD= 000000  FD.DIR= 000010
CHAN07  001604R  025    CHR..U= 000165  DEVIL   005114R  025 D.QUAD= 000000  FD.FTN= 000001
CHAN08  001766R  025    CHR..Z= 000172  DEVNTF= 104470  D.RES = 000000  FD.F11= 040000
CHAN09  002150R  025    CHR..0= 000060  DIGITS  000006R  025 D.RHO = 000000  FD.INS= 000010
CHAN10  002332R  025    CHR..1= 000061  DIMAX  = 000020  D.ROT = 000003  FD.ISP= 002000
CHAN11  002514R  025    CH.AND= 000001  DIMBLK  003112R  025 D.SCON= 000006  FD.MNT= 100000
CHAN12  002676R  025    CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006  FD.OSP= 004000
CHAN13  000022R  023    CMDMAX= 000010  DIMTYP  003116R  025 D.SUB = 000000  FD.PLC= 000004
CHAN14  000204R  023    CNTSIZ  005130R  025 DIRFUL= 104466  D.SUBS= 000006  FD.PRN= 000004
CHNOPN= 104471          COMBIN  005076R  025 DOMERR= 104417  D.TAKE= 000002  FD.PSE= 010000
CHRAND= 000051          COMCNT  005126R  025 DPTERR= 104416  D.TRAN= 000004  FD.RAH= 000001
CHRBEL= 000007          COMERR= 104425  DYDCNT  003224R  025 D.UNON= 000006  FD.RAN= 000002
CHRBKR= 000133          COMPTR  005122R  025 D.AND  = 000000  D.$MAX= 000000  FD.REC= 000001
CHRBOL= 000200          COMRTN  003156R  025 D.BXCT= 000007  D.$MIN= 000000  FD.RWM= 000001
CHRBOX= 000114          COMSKP  005124R  025 D.BXIO= 000007  D.$MUL= 000000  FD.SDI= 000020
CHRCMA= 000054          CORPTR  005120R  025 D.BXNL= 000007  D.$PLU= 000000  FD.SQD= 000040
CHRCOL= 000076          CS.ASS= 000200  D.BXPP= 000007  D.$RES= 000000  FD.TTY= 000004
CHRDEL= 000107          CS.BOO= 000010  D.BXPW= 000007  D.$SUB= 000000  FD.WBH= 000002
CHRDIV= 000053          CS.FLO= 000020  D.BXRL= 000007  D.1COM= 000001  FF.CHR= 000005
CHRDLR= 000176          CS.IDN= 000040  D.CIRC= 000001  D.1EXP= 000002  FF.NV  = 000003
CHRDLT= 000110          CS.ONE= 000100  D.COMB= 000001  D.1ROT= 000003  FF.POE= 000002
CHRDOT= 000056          CTEPS   000152R  025 D.COMP= 000001  EDICHR  005063R  025 FF.RWD= 000001
CHREP  = 000105         CTMAX   000212R  025 D.CONT= 000006  EDICNT  005064R  025 FF.RWF= 000006
CHRERR= 104431          CTRMAX  000172R  025 D.CTLM= 000000  EDIFLG  005062R  025 FF.SPC= 000004
CHRFIL= 000177          C.ALOC= 000162  D.DEAL= 000003  EDIT   = 000003  FHEADR  005012R  025
CHRFRO= 000135          C.BLOK= 000064  D.DECD= 000002  EIS    = 000000  FHRET   005040R  025
```

EXISTS MACRO M1113 08-FEB-80 12:11 PAGE 46-2
SYMBOL TABLE

```
FILNTF= 104467          F.FNB = 000102          ILFNAM= 104464          MONSFT  003126R 025 M.SCAN= 000011
FIS   = 000000          F.FNDF= 000010          ILLCHN= 104540          MONTYP  003124R 025 M.SHAP= 000000
FISCON= 046000          F.FTYP= 000116          ILLSW = 104565          MONVAL  003134R 025 M.SIGN= 000000
FIXPTR  003074R 025     F.FUN = 000010          INDERR= 104410          M.ABS = 000000      M.SUBR= 000001
FLAGS   004764R 025     F.FVER= 000120          INNTMP  003250R 025     M.BOL = 000200      M.SYM = 000203
FLCBUF  003250R 025     F.HEAD= 000006          INNTM2  005016R 025     M.BXAV= 000007      M.SYN = 000201
FLINE   005014R 025     F.HIBK= 000004          INPTR   000064R 025     M.BXCR= 000007      M.TRAN= 000012
FLOEXP  003270R 025     F.LBL = 000020          INSNAM= 104462          M.BXCT= 000007      M.$ABS= 000000
FLONEG  003272R 025     F.LINE= 000012          IOERR = 104477          M.BXEX= 000007      M.$CEI= 000000
FLSWTB  000222R 025     F.LOC = 000004          I.CNT = 177776          M.BXFX= 000007      M.$FLR= 000000
FLTLEN= 000010          F.LSTC= 000004          I.DIM = 177772          M.BXIO= 000007      M.$IDN= 000000
FNINSI  005044R 025     F.LUN = 000042          I.LEN = 000012          M.BXLC= 000003      M.$NEG= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.OERR= 000062          M.BXNC= 000007      M.$SIG= 000000
FO.APD= 000106          F.MBC1= 000055          I.OLD = 177770          M.BXNL= 000007      M.1RDT= 000011
FO.MFY= 000002          F.MBFG= 000056          I.PTR = 177774          M.BXPP= 000007      M.1REV= 000003
FO.RD = 000001          F.MON = 000002          I.STR = 177766          M.BXPW= 000007      M.1SCN= 000011
FO.UPD= 000006          F.NIL = 000001          LABERR= 104406          M.BXRL= 000007      NAMBUF  003240R 025
FO.WRT= 000016          F.NRBD= 000024          LABLPT  005042R 025     M.BXTS= 000007      NB.DEV= 000200
FPMODE  000042R 025     F.NREC= 000030          LARGBL  003162R 025     M.BXWA= 000007      NB.DIR= 000100
FPP   = 000000          F.OPN = 000010          LARGCV  003176R 025     M.CEI = 177777      NB.NAM= 000004
FPPLEN= 000004          F.OVBS= 000030          LARGFL  003160R 025     M.CHR = 000202      NB.SD1= 000400
FREBLK  007400R 025     F.PREV= 000000          LARGLE  003174R 025     M.CMT = 000206      NB.SD2= 001000
FUZZ    000142R 025     F.PRNT= 100000          LARGLN  003166R 025     M.DOUT= 000006      NB.SNM= 000040
FXPTR1  003076R 025     F.PROT= 010000          LARGPT  003164R 025     M.DQAD= 000006      NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025     M.ENCD= 000004      NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025     M.EXE = 000000      NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025     M.EXP = 000001      NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025     M.FACT= 000001      ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025     M.FIL = 000177      ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621          M.FLR = 177777      ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025     M.FMT = 000004      ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412          M.GDUP= 000003      ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025     M.GDWN= 000003      ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025     M.GETI= 000006      ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025     M.IBM = 000003      ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025     M.IDN = 000000      ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025     M.INSP= 000006      NEGSGN  005063R 025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025     M.IOTA= 000000      NINPT1  003302R 025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020          M.LIT = 000205      NINPT2  003300R 025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025     M.LN  = 000001      NINSTK  003304R 025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541          M.MMD = 000010      NINTMP  003310R 025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025     M.NEG = 000000      NINTYP  003306R 025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025     M.NOT = 000000      NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025     M.NUB = 000006      NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025     M.NUM = 000204      NPRT1   004762R 025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012      NULL    007364R 025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000      NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006      NUMDFL  003230R 025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000      NUMEFL  003233R 025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000      NUMESN  003236R 025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012      NUMFLG  003230R 025
F.ERR = 000052          GNOME   003102R 025     MACHIN= 000106          M.RAVL= 000000      NUMLEN  000062R 025
F.FACC= 000043          HASHC   005136R 025     MAXCHN= 000014          M.RCIP= 000000      NUMNEG  003232R 025
F.FFBY= 000014          HASHTB  005140R 025     MONBLK  003120R 025     M.RDCT= 000011      NUMQFL  003234R 025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R 025     M.REVS= 000003      NUMRIG  003231R 025
F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R 025     M.ROLL= 000000      NUMWID  003226R 025
F.FNAM= 000110          ICHAN   005050R 025     MONPTR  003122R 025     M.SATA= 000001      NUMZFL  003235R 025
```

SYMBOL TABLE

```
NYIERR= 104432           RARGCV  003220R     025 R.PREV= 000004          SY.ERP= 000020          T.MON = 000010
N.BLSZ= 000036           RARGFL  003202R     025 R.PTR = 000000          SY.NOU= 000100          T.NIL = 000004
N.DESC= 000020           RARGLE  003216R     025 R.SEQ = 000003          SY.OPN= 000002          T.NUM = 002000
N.DEV = 000032           RARGLN  003210R     025 R.VAR = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034           RARGPT  003206R     025 R6    =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DID = 000024           RARGSF  003214R     025 R7    =%000007          SY.SVC= 000004          T.SOP = 000002
N.DVNM= 000032           RARGTY  003212R     025 SI    =%000005          SY.SVL= 000200          T.SPOP= 000040
N.EXT = 000014           RARGVA  003222R     025 SIDAMN= 104433          S.AB  = 177776          T.TMP = 000200
N.FID = 000000           RBASE   005074R     025 SIFRAM  003314R   025 S.ASB = 100000          T.VAL = 100000
N.FNAM= 000006           RCOUNR  005112R     025 SISTAK  006350R   025 S.BFHD= 000020          T.VAR = 001000
N.FTYP= 000014           RCOUNT  005110R     025 SITOP   006346R   025 S.BITS= 000002          UPPER = 177400
N.FVER= 000016           REDCNT  005070R     025 SITP    007144R   025 S.CHR1= 000003          UTNAME  000344R     023
N.NAME= 000006           REDIGN  005066R     025 SIWARN= 000200          S.FATT= 000016          VALERR= 104413
N.NEXT= 000022           REDOFF  005072R     025 SIZTAB  000044R   025 S.FDB = 000140          VARBLE  007400R     025
N.PPN = 000024           REENTR  003060R     025 SI.BIT= 177774          S.FLB = 040000          VARBLS  000000R     025
N.STAT= 000020           RESBIT  003142R     025 SI.LEN= 177772          S.FNAM= 000006          VERNUM  000000R     025
N.UNIT= 000034           RESCHN= ****** GX       SI.RET= 177772          S.FNB = 000036          VRSION= 000002
OBUFLN= 001000           RESCVT  003152R     025 SI.VAL= 177776          S.FNBW= 000017          WIDTH   000072R     025
OCHAN   005052R     025  RESERR= 104440          SKPSIZ  005134R   025 S.FNTY= 000004          WROUND  000020R     025
OMODE   000402R     023  RESLEN  003150R     025 SPHOLD  000004R   024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP0  005066R     025  RESPTR  003140R     025 SPINIT  000002R   024 S.LEN = 000002          WSFULL= 104403
OPTMP1  005070R     025  RESSFT  003146R     025 STKINC= 000774          S.LINK= 177774          WSNAME  000074R     025
OPTMP2  005072R     025  RESTYP  003144R     025 STOP  = 104420          S.NFEN= 000020          WSTIME  000132R     025
OPTMP3  005074R     025  RESULT  003136R     025 SUBCNT  003104R   025 S.TYPE= 140000          $FIS  = ****** G
OPTMP4  005076R     025  RESVAL  003154R     025 SUBDCT  003106R   025 S.VAL = 000000          $FPP  = ****** G
OPTMP5  005100R     025  RINC    005104R     025 SUBPTR  003110R   025 TKDIMS  005072R   025 $FPPLN= ****** G
OPTMP6  005102R     025  RLTBAS  005066R     025 SYMB1   005344R   025 TKFRAM  005066R   025 $MACHN= ****** G
OPTMP7  005104R     025  ROTINC  005070R     025 SYMF    005340R   025 TKLEFT  005070R   025 $RSTS = ****** G
OPTMP8  005106R     025  ROUND   000010R     025 SYMPT1  005346R   025 TMODE   000403R   023 $RSX  = ****** G
OPTMP9  005110R     025  ROVER   003312R     025 SYMTAB  005352R   025 TMPTAG= 000166       $RT11 = ****** G
OPTM10  005112R     025  RSTS  = 000000          SYMTBF= 104404          TTYBUF  004140R   025 .BOOLE= 000002
OPTM14  005026R     025  RSX   = 000115          SYMTOP  005350R   025 TTYLEN= 000622          .CHARA= 000000
ORIGIN  000030R     025  RTOP    005070R     025 SYMTP   006272R   025 TURKEY= 104436          .DEF. = 000002
OUTPTR  000066R     025  RT11  = 000000          SYMWRN= 000054          TYPE  = 000166          .FIND = ****** GX
OVROPR= 000000           R.BITS= 000003          SYNERR= 104407          T.BOL = 020000          .FLOAT= 000006
PATCH   007344R     025  R.FIX = 000001          SYSERR= 104402          T.CON = 004000          .FSRCB= ****** G
PRECSN  000052R     025  R.LCT = 000002          SYSTAT  000000R   024 T.DIM = 000100          .INTEG= 000004
RANDAC  003064R     025  R.LN  = 000002          SY.BAD= 000001          T.FUN = 010000          .MOLUN  000404R     023
RANDY   003062R     025  R.LNUM= 000010          SY.BRK= 100000          T.LIST= 000400          ...GBL= 000000
RANERR= 104411           R.NEXT= 000006          SY.BS = 000010          T.LIT = 000020          ...TPC= 000020
RARGBL  003204R     025  R.OFF = 000000          SY.CLR= 000040

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
EXISTS  MACRO M1113  08-FEB-80 12:11  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000024      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12898 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:54
EXISTS,DM1:[305,301]EXISTS/-SP=APLRX1,APLDEF,EXISTS,END
```

```
**FILE**ID**EXITIT
```

```
EEEEEEEEEE   XX      XX     IIIIII     TTTTTTTTTT     IIIIII     TTTTTTTTTT
EEEEEEEEEE   XX      XX     IIIIII     TTTTTTTTTT     IIIIII     TTTTTTTTTT
EE           XX      XX       II         TT           II         TT
EE           XX      XX       II         TT           II         TT
EE             XX  XX         II         TT           II         TT
EE             XX  XX         II         TT           II         TT
EEEEEEEE         XX           II         TT           II         TT
EEEEEEEE         XX           II         TT           II         TT
EE             XX  XX         II         TT           II         TT
EE             XX  XX         II         TT           II         TT
EE           XX      XX       II         TT           II         TT          ....
EE           XX      XX       II         TT           II         TT          ....
EEEEEEEEEE   XX      XX     IIIIII       TT         IIIIII       TT          ....
EEEEEEEEEE   XX      XX     IIIIII       TT         IIIIII       TT          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:13
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS      OPERATOR OVERLAYS
     5-   268    DEFINITIONS      REGISTERS
     6-   331    DEFINITIONS      MEMORY ORGANIZATION
     7-   377    DEFINITIONS      ORDERING PSECTS
     8-   426    DEFINITIONS      SI PARSE FRAMES
     9-   474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS      SI EXECUTION FRAMES
    11-   580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS      CHRTAB BITS
    12-   624    DEFINITIONS      SYSTEM STATUS BITS
    13-   640    DEFINITIONS      F-FRAMES
    14-   724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS      LIST STORAGE BLOCK
    18-   911    DEFINITIONS      INDEXING STACK FRAME
    19-   940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS      CHANNEL BLOCK
    22-  1085    DEFINITIONS              RSX DATA
    23-  1110    DEFINITIONS      NAME BLOCKS
    33-  1612    DEFINITIONS      MACROS
    35-  1905    DEFINITIONS      APL CHARACTERS
    36-  1967    DEFINITIONS      ERRORS
    44-     4    EXITIT           TERMINATE APL SESSION
```

```
EXITIT  MACRO M1113  08-FEB-80 12:13  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  EXITIT
                                .IDENT  \V02.03\
                                .SBTTL  EXITIT          TERMINATE APL SESSION
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'EXITIT' cleans up the tty for exiting then exits
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       FRESET  - Close all files
                                ;       TTOUT   - Tramsmit character to terminal (untranslated)
                                ;       WRITEB  - PRINT OUT ALL THAT GOOD STUFF
                                ;
                                ; SIDE EFFECTS:
                                ;       An obvious one: APL terminates.
                                ;--

000000                          .PSECT  CODE

000000                          EXITIT::
000000                                  CALL    FRESET          ;CLOSE ALL FILES
000004  012737  000256' 005052'         MOV     #CHAN00,OCHAN   ;SET OUTPUT CHANNEL
000012  105737  000403'                 TSTB    TMODE           ;WHAT MODE ARE WE IN ?
000016  001004                          BNE     DIE             ;---TTY MODE, NO ESCAPE SEQUENCE
000020  012700  000017                  MOV     #17,R0          ;^O
000024                                  CALL    TTOUT           ;PRINT IT
000030                          DIE::
000030                                  EXIT$S                  ;GOODBYE, DORATHY, GOODBYE
000030  012746                          MOV     (PC)+,-(SP)
000032     063  001                    .BYTE   51.,1
000034  104377                          EMT     ^O<377>
```

```
EXITIT  MACRO M1113  08-FEB-80 12:13  PAGE 46
EXITIT           TERMINATE APL SESSION

      1          000001                              .END
```

EXITIT MACRO M1113 08-FEB-80 12:13 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DECD= 000002    EIS    = 000000
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DIV = 000000    ELESIZ 005132R     025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DMD = 000010    ENDCOR 000004R     025
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DQAD= 177777    EOF    = 104475
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.DROP= 000002    ERRMAX= 000166
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.ENCD= 000002    ERRNUM 004772R     025
ALLINT  003274R     025  CHRPLU= 000055    C.DIG = 000040    D.EQ  = 000000    ERRPTR 004770R     025
ARYRET  003276R     025  CHRPWR= 000120    C.FASW= 020000    D.EXPD= 000002    ERR0   004774R     025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FGET= 000005    EXEBUF 005054R     025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FMT = 000004    EXITIT 000000RG    020
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPNT= 000005    E$CT   005056R     025
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.FPUT= 000005    E$PT   005060R     025
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GE  = 000000    FA.APD= 000100
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.GT  = 000000    FA.CRE= 000010
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IBM = 000002    FA.DLK= 001000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.IND = 000002    FA.ENB= 100000
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INNR= 000013    FA.EXC= 002000
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.INTR= 000006    FA.EXT= 000004
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LE  = 000000    FA.NSP= 000100
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LOG = 000001    FA.NSR= 000200
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.LT  = 000000    FA.POS= 010000
BLZBUB  005116R     025  CHR.LF= 000012    C.SPFX= 000100    D.MAX = 000000    FA.RD = 000001
BRSELF  005046R     025  CHR.LT= 000043    C.SPGL= 000200    D.MEM = 000002    FA.RWD= 004000
CCNT    005072R     025  CHR.SI= 000017    C.SPLX= 000001    D.MIN = 000000    FA.SEQ= 040000
CHANEX  003060R     025  CHR.SL= 000057    C.SPRS= 000002    D.MUL = 000000    FA.SHR= 000040
CHAN00  000256R     025  CHR.SO= 000016    C.SRC = 177774    D.NE  = 000000    FA.TMP= 000020
CHAN01  000330R     025  CHR.TB= 000011    C.SRCP= 000024    D.NN  = 000000    FA.WCK= 020000
CHAN02  000512R     025  CHR.UL= 000106    C.TT  = 000004    D.NR  = 000000    FA.WRT= 000002
CHAN03  000674R     025  CHR..A= 000141    C.WIDC= 177770    D.OR  = 000000    FD.BLK= 000010
CHAN04  001056R     025  CHR..E= 000145    C.WIDR= 177772    D.PLUS= 000000    FD.CCL= 000002
CHAN05  001240R     025  CHR..S= 000163    DEBUG = 000000    D.PSUB= 000006    FD.COM= 020000
CHAN06  001422R     025  CHR..T= 000164    DEFERR= 104405    D.PWR = 000001    FD.CR = 000002
CHAN07  001604R     025  CHR..U= 000165    DEVIL   005114R   025 D.QQAD= 000000    FD.DIR= 000010
CHAN08  001766R     025  CHR..Z= 000172    DEVNTF= 104470    D.QUAD= 000000    FD.FTN= 000001
CHAN09  002150R     025  CHR..0= 000060    DIE     000030RG  020 D.RES = 000000    FD.F11= 040000
CHAN10  002332R     025  CHR..1= 000061    DIGITS  000006R   025 D.RH0 = 000000    FD.INS= 000010
CHAN11  002514R     025  CH.AND= 000001    DIMAX = 000020        D.ROT = 000003    FD.ISP= 002000
CHAN12  002676R     025  CMDBUF  003250R   025 DIMBLK  003112R   025 D.SCON= 000006    FD.MNT= 100000
CHAN13  000022R     023  CMDMAX= 000010        DIMPTR  003114R   025 D.SDIF= 000006    FD.OSP= 004000
CHAN14  000204R     023  CNTSIZ  005130R   025 DIMTYP  003116R   025 D.SUB = 000000    FD.PLC= 000004
CHNOPN= 104471           COMBIN  005076R   025 DIRFUL= 104466        D.SUBS= 000006    FD.PRN= 000004
CHRAND= 000051           COMCNT  005126R   025 DOMERR= 104417        D.TAKE= 000002    FD.PSE= 010000
CHRBEL= 000007           COMERR= 104425        DPTERR= 104416        D.TRAN= 000004    FD.RAH= 000001
CHRBKR= 000133           COMPTR  005122R   025 DYDCNT  003224R   025 D.UNON= 000006    FD.RAN= 000002
CHRBOL= 000200           COMRTN  003156R   025 D.AND = 000000        D.$MAX= 000000    FD.REC= 000001
CHRBOX= 000114           COMSKP  005124R   025 D.BXCT= 000007        D.$MIN= 000000    FD.RWM= 000001
CHRCMA= 000054           CORPTR  005120R   025 D.BXIO= 000007        D.$MUL= 000000    FD.SDI= 000020
CHRCOL= 000076           CS.ASS= 000200        D.BXNL= 000007        D.$PLU= 000000    FD.SQD= 000040
CHRDEL= 000107           CS.BOO= 000010        D.BXPP= 000007        D.$RES= 000000    FD.TTY= 000004
CHRDIV= 000053           CS.FLO= 000020        D.BXPW= 000007        D.$SUB= 000000    FD.WBH= 000002
CHRDLR= 000176           CS.IDN= 000040        D.BXRL= 000007        D.1COM= 000001    FF.CHR= 000005
CHRDLT= 000110           CS.ONE= 000100        D.CIRC= 000001        D.1EXP= 000002    FF.NV = 000003
CHRDOT= 000056           CTEPS   000152R   025 D.COMB= 000001        D.1ROT= 000003    FF.POE= 000002
CHREP = 000105           CTMAX   000212R   025 D.COMP= 000001        EDICHR  005063R   025 FF.RWD= 000001
CHRERR= 104431           CTRMAX  000172R   025 D.CONT= 000006        EDICNT  005064R   025 FF.RWF= 000006
CHRFIL= 000177           C.ALOC= 000162        D.CTLM= 000000        EDIFLG  005062R   025 FF.SPC= 000004
CHRFRO= 000135           C.BLOK= 000064        D.DEAL= 000003        EDIT   = 000003       FHEADR  005012R     025
```

EXITIT MACRO M1113 08-FEB-80 12:13 PAGE 46-2
SYMBOL TABLE

```
FHRET   005040R     025 F.FLB = 000100          HSHSIZ= 000200          MONLEN  003130R     025 M.ROLL= 000000
FILNTF= 104467          F.FNAM= 000110          ICHAN   005050R 025 MONPTR  003122R     025 M.SATA= 000001
FIS   = 000000          F.FNB = 000102          ILFNAM= 104464          MONSFT  003126R     025 M.SCAN= 000011
FISCON= 046000          F.FNDF= 000010          ILLCHN= 104540          MONTYP  003124R     025 M.SHAP= 000000
FIXPTR  003074R     025 F.FTYP= 000116          ILLSW = 104565          MONVAL  003134R     025 M.SIGN= 000000
FLAGS   004764R     025 F.FUN = 000010          INDERR= 104410          M.ABS = 000000          M.SUBR= 000001
FLCBUF  003250R     025 F.FVER= 000120          INNTMP  003250R 025 M.BOL = 000200          M.SYM = 000203
FLINE   005014R     025 F.HEAD= 000006          INNTM2  005016R 025 M.BXAV= 000007          M.SYN = 000201
FLOEXP  003270R     025 F.HIBK= 000004          INPTR   000064R 025 M.BXCR= 000007          M.TRAN= 000012
FLONEG  003272R     025 F.LBL = 000020          INSNAM= 104462          M.BXCT= 000007          M.$ABS= 000000
FLSWTB  000222R     025 F.LINE= 000012          IOERR = 104477          M.BXEX= 000007          M.$CEI= 000000
FLTLEN= 000010          F.LOC = 000004          I.CNT = 177776          M.BXFX= 000007          M.$FLR= 000000
FNINSI  005044R     025 F.LSTC= 000004          I.DIM = 177772          M.BXIO= 000007          M.$IDN= 000000
FORIGN  000032R     025 F.LUN = 000042          I.LEN = 000012          M.BXLC= 000003          M.$NEG= 000000
FO.APD= 000106          F.MBCT= 000054          I.OERR= 000062          M.BXNC= 000007          M.$SIG= 000000
FO.MFY= 000002          F.MBC1= 000055          I.OLD = 177770          M.BXNL= 000007          M.1RDT= 000011
FO.RD = 000001          F.MBFG= 000056          I.PTR = 177774          M.BXPP= 000007          M.1REV= 000003
FO.UPD= 000006          F.MON = 000002          I.STR = 177766          M.BXPW= 000007          M.1SCN= 000011
FO.WRT= 000016          F.NIL = 000001          LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R     025
FPMODE  000042R     025 F.NRBD= 000024          LABLPT  005042R 025 M.BXTS= 000007          NB.DEV= 000200
FPP   = 000000          F.NREC= 000030          LARGBL  003162R 025 M.BXWA= 000007          NB.DIR= 000100
FPPLEN= 000004          F.OPN = 000010          LARGCV  003176R 025 M.CEI = 177777          NB.NAM= 000004
FREBLK  007400R     025 F.OVBS= 000030          LARGFL  003160R 025 M.CHR = 000202          NB.SD1= 000400
FRESET= ****** GX       F.PREV= 000000          LARGLE  003174R 025 M.CMT = 000206          NB.SD2= 001000
FUZZ    000142R     025 F.PRNT= 100000          LARGLN  003166R 025 M.DOUT= 000006          NB.SNM= 000040
FXPTR1  003076R     025 F.PROT= 010000          LARGPT  003164R 025 M.DQAD= 000006          NB.STP= 000020
F$EVAL= 000100          F.RACC= 000016          LARGSF  003172R 025 M.ENCD= 000004          NB.SVR= 000010
F$EVI = 000200          F.RATT= 000001          LARGTY  003170R 025 M.EXE = 000000          NB.TYP= 000002
F$EXEC= 000400          F.RCNM= 000034          LARGVA  003200R 025 M.EXP = 000001          NB.VER= 000001
F$FUNC= 001000          F.RCTL= 000017          LBINC   005100R 025 M.FACT= 000001          ND.DEV= 000200
F.ACTL= 000076          F.READ= 004000          LBUFF   003316R 025 M.FIL = 000177          ND.EXT= 000002
F.ALOC= 000040          F.REDF= 000040          LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BBFS= 000062          F.RES = 000040          LCOUNT  005106R 025 M.FMT = 000004          ND.PPN= 000100
F.BDB = 000070          F.RET = 177772          LENERR= 104412          M.GDUP= 000003          ND.$EX= 000000
F.BGBC= 000057          F.RSIZ= 000002          LENTAB  000054R 025 M.GDWN= 000003          ND.$E$= 000020
F.BITS= 177774          F.RTYP= 000000          LEXPTR  003100R 025 M.GETI= 000006          ND.$NM= 000000
F.BKDN= 000026          F.R2  = 000014          LFTOP   005066R 025 M.IBM = 000003          ND.$N$= 000040
F.BKDS= 000020          F.R3  = 000016          LINC    005102R 025 M.IDN = 000000          ND.$PN= 001400
F.BKEF= 000050          F.R4  = 000020          LINE    005016R 025 M.INSP= 000006          NEGSGN  005063R     025
F.BKP1= 000051          F.SEQN= 000100          LININC  005026R 025 M.IOTA= 000000          NINPT1  003302R     025
F.BKST= 000024          F.SPDV= 000072          LISIZE= 000020          M.LIT = 000205          NINPT2  003300R     025
F.BKVB= 000064          F.SPUN= 000074          LITOP   005072R 025 M.LN  = 000001          NINSTK  003304R     025
F.BR  = 000020          F.STBK= 000036          LNLNG = 104541          M.MMD = 000010          NINTMP  003310R     025
F.CHR = 000075          F.STK = 000022          LOCLCT  005036R 025 M.NEG = 000000          NINTYP  003306R     025
F.CNTG= 000034          F.STOP= 000004          LOWCOR  000002R 025 M.NOT = 000000          NOTAVL= 104472
F.CODE= 000010          F.STP = 000001          LSTERP  004766R 025 M.NUB = 000006          NOTOPN= 104473
F.DFNB= 000046          F.SUSP= 000002          LWIDTH  000070R 025 M.NUM = 000204          NPRT1   004762R     025
F.DSPT= 000044          F.TRC = 000002          L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.DVNM= 000134          F.UNIT= 000136          L.CNT = 000002          M.PITM= 000000          NUMCHN= 000020
F.EFBK= 000010          F.URBD= 000020          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.EFN = 000050          F.VAL = 177776          L.PTR = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.END = 002000          F.VBN = 000064          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.EOBB= 000032          F.VBSZ= 000060          L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.ERR = 000052          GNOME   003102R 025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FACC= 000043          HASHC   005136R 025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
F.FFBY= 000014          HASHTB  005140R 025 MONBLK  003120R 025 M.RDCT= 000011          NUMQFL  003234R     025
F.FLAG= 000002          HSHMSK= 177601          MONCVT  003132R 025 M.REVS= 000003          NUMRIG  003231R     025
```

```
EXITIT  MACRO M1113  08-FEB-80 12:13  PAGE 46-3
SYMBOL TABLE

NUMWID  003226R     025 RANERR= 104411          R.OFF  = 000000          SY.CLR= 000040          T.LIT  = 000020
NUMZFL  003235R     025 RARGBL  003204R     025 R.PREV= 000004          SY.ERP= 000020          T.MON  = 000010
NYIERR= 104432          RARGCV  003220R     025 R.PTR  = 000000          SY.NOU= 000100          T.NIL  = 000004
N.BLSZ= 000036          RARGFL  003202R     025 R.SEQ  = 000003          SY.OPN= 000002          T.NUM  = 002000
N.DESC= 000020          RARGLE  003216R     025 R.VAR  = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEV = 000032          RARGLN  003210R     025 R6     =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DEVN= 000034          RARGPT  003206R     025 R7     =%000007          SY.SVC= 000004          T.SOP  = 000002
N.DID = 000024          RARGSF  003214R     025 SI     =%000005          SY.SVL= 000200          T.SPOP= 000040
N.DVNM= 000032          RARGTY  003212R     025 SIDAMN= 104433           S.AB   = 177776          T.TMP  = 000200
N.EXT = 000014          RARGVA  003222R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAL  = 100000
N.FID = 000000          RBASE   005074R     025 SISTAK  006350R     025 S.BFHD= 000020          T.VAR  = 001000
N.FNAM= 000006          RCOUNR  005112R     025 SITOP   006346R     025 S.BITS= 000002          UPPER  = 177400
N.FTYP= 000014          RCOUNT  005110R     025 SITP    007144R     025 S.CHR1= 000003          UTNAME  000344R     023
N.FVER= 000016          REDCNT  005070R     025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.NAME= 000006          REDIGN  005066R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLE  007400R     025
N.NEXT= 000022          REDOFF  005072R     025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R     025
N.PPN  = 000024         REENTR  003060R     025 SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R     025
N.STAT= 000020          RESBIT  003142R     025 SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
N.UNIT= 000034          RESCVT  003152R     025 SI.VAL= 177776           S.FNBW= 000017          WIDTH   000072R     025
OBUFLN= 001000          RESERR= 104440          SKPSIZ  005134R     025 S.FNTY= 000004          WROUND  000020R     025
OCHAN   005052R     025 RESLEN  003150R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSDATE  000136R     025
OMODE   000402R     023 RESPTR  003140R     025 SPINIT  000002R     024 S.LEN  = 000002          WSFULL= 104403
OPTMP0  005066R     025 RESSFT  003146R     025 STKINC= 000774           S.LINK= 177774          WSNAME  000074R     025
OPTMP1  005070R     025 RESTYP  003144R     025 STOP   = 104420          S.NFEN= 000020          WSTIME  000132R     025
OPTMP2  005072R     025 RESULT  003136R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP3  005074R     025 RESVAL  003154R     025 SUBDCT  003106R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP4  005076R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP6  005102R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP7  005104R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSX   = ****** G
OPTMP8  005106R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166          $RT11  = ****** G
OPTMP9  005110R     025 RSTS   = 000000          SYMTBF= 104404           TTOUT  = ****** GX      .BOOLE= 000002
OPTM10  005112R     025 RSX    = 000115          SYMTOP  005350R     025 TTYBUF  004140R     025 .CHARA= 000000
OPTM14  005026R     025 RTOP    005070R     025 SYMTP   006272R     025 TTYLEN= 000622          .DEF.  = 000002
ORIGIN  000030R     025 RT11   = 000000          SYMWRN= 000054           TURKEY= 104436          .FLOAT= 000006
OUTPTR  000066R     025 R.BITS= 000003           SYNERR= 104407           TYPE   = 000166          .FSRCB= ****** G
OVROPR= 000000          R.FIX  = 000001          SYSERR= 104402           T.BOL  = 020000          .INTEG= 000004
PATCH   007344R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.CON  = 004000          .MOLUN  000404R     023
PRECSN  000052R     025 R.LN   = 000002          SY.BAD= 000001           T.DIM  = 000100          ...GBL= 000000
RANDAC  003064R     025 R.LNUM= 000010           SY.BRK= 100000           T.FUN  = 010000          ...TPC= 000020
RANDY   003062R     025 R.NEXT= 000006           SY.BS  = 000010          T.LIST= 000400

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
MNTBL   000000  016
```

```
EXITIT  MACRO M1113  08-FEB-80 12:13  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000     017
CODE    000036     020
PATCH   000000     021
$$FSR1  000000     022
DEF$0   000406     023
DEF$1   000006     024
DEF$2   007400     025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12904 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:01:57
EXITIT,DM1:[305,301]EXITIT/-SP=APLRX1,APLDEF,EXITIT,END
```

```
**FILE**ID**EXPR
```

```
EEEEEEEEEE    XX      XX    PPPPPPPP      RRRRRRRR
EEEEEEEEEE    XX      XX    PPPPPPPP      RRRRRRRR
EE            XX      XX    PP      PP    RR      RR
EE            XX      XX    PP      PP    RR      RR
EE              XX  XX      PP      PP    RR      RR
EE              XX  XX      PP      PP    RR      RR
EEEEEEEE          XX        PPPPPPPP      RRRRRRRR
EEEEEEEE          XX        PPPPPPPP      RRRRRRRR
EE              XX  XX      PP            RR  RR
EE              XX  XX      PP            RR  RR
EE            XX      XX    PP            RR    RR          ....
EE            XX      XX    PP            RR    RR          ....
EEEEEEEEEE    XX      XX    PP            RR      RR        ....
EEEEEEEEEE    XX      XX    PP            RR      RR        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 12:15
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    EXPR             EVALUTE APL EXPRESSION
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 1


      1        000000                 RT11    =  0          ;NOT RT-11
      2        000000                 RSTS    =  0          ;NOT RSTS/E
      3        000115                 RSX     = 'M          ;RSX-11/M
      4
      5        000000                 EIS     =  0          ;NO EIS
      6        000000                 FIS     =  0          ;NO FIS
      7        000000                 FPP     =  0          ;NO FPP
      8
      9        000004                 FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                 DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                                .TITLE  EXPR
                                .IDENT  \V02.03\

                                .SBTTL  EXPR            EVALUTE APL EXPRESSION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       this is pass 3 of line interpretation.  Tokens in  the
                                ;       code string produced by pass 2 are processsed by calling
                                ;       operators and functions according to apl syntax rules.
                                ;       Evaluation is easy except for a special case described below.
                                ;       A line is evaluated until some non-statement token is encountered.
                                ;       This token is passed back to the caller, and it is his res-
                                ;       ponsibility to determine if it is a bol or keyword or the like.
                                ;       This routine must be passed the last token in the line
                                ;       for initialization.
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    EXPR
                                ;
                                ; INPUTS:
                                ;       R2      - Pointer to the codestring
                                ;
                                ; IMPLICIT INPUTS:
                                ;       SI      - The SI contains the "stack" for expressions, ...
                                ;
                                ; OUTPUTS:
                                ;       R2      - Pointer to end of evaluated codestring
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       SI      - The SI contains the result of EXPR as the top frame.
                                ;
                                ; ROUTINES CALLED:
                                ;       VAL     - Used to get the next value in the codestring
                                ;       WORKER  - (A secondary call of EXPR)
                                ;       BRKCHK  - Checks for CNTL-C interrupts
                                ;       GETNXT  - gets the next token in the codestring,
                                ;                 places the info in registers
                                ;       SPASSN  - Used for 'special' (i.e., non-variable) assignments
                                ;       ASSIGN  - Used for more orthodox assignments
                                ;       EXPR    - (A little recursion here)
                                ;       DOT     - Checks syntax for inner, outer products
                                ;       MONAD   - Called for all monadic operations
                                ;       DYAD    - Called for all dyadic operations
                                ;
                                ; SIDE EFFECTS:
                                ;       This routine evaluates a codestring and returns
                                ;       the result on the SI
                                ;--
000000                                  .PSECT CODE


000000  012625          EXPR::  MOV     (SP)+,(SI)+     ;PUSH RETURN ON SI
000002  005025                  CLR     (SI)+           ;CLEAR BITS
000004  005025                  CLR     (SI)+           ;CLEAR VALUE
000006                          CALL    VAL             ;GET INITIAL VALUE
000012  000407                  BR      1$              ;WHAT? NO VAL?
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 44-1
EXPR            EVALUTE APL EXPRESSION

     58 000014  010500                          MOV     SI,R0           ;COPY POINTER
     59 000016  062700  177772                  ADD     #SI.LEN,R0      ;BACK UP A FRAME
     60 000022  014540                          MOV     -(SI),-(R0)     ;MOVE VALUE
     61 000024  014540                          MOV     -(SI),-(R0)     ;AND COPY BITS
     62 000026  005045                          CLR     -(SI)           ;CLEAR RETURN
     63 000030  000410                          BR      WORKER          ;AND START THE PRESSES!
     64
     65                                 ;CHECK FOR BLANK LINE OR OTHER NONSENSE
     66
     67 000032  032765  000400  177774  1$:     BIT     #T.LIST,SI.BIT(SI);LIST ON SI?
     68 000040  001154                          BNE     EXPSYN          ;YES SYNTAX ERROR
     69 000042  062705  177772                  ADD     #SI.LEN,SI      ;POP THE VAL FRAME
     70 000046  000175  177772                  JMP     @SI.RET(SI)     ;ELSE JUST RETURN
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 45
EXPR            EVALUTE APL EXPRESSION


                                        ;+
                                        ; START OF MAIN LOOP.  THIS ROUTINE CALLS ALL MONADIC AND
                                        ; DYADIC OPS AND FUNCTIONS
                                        ;-
000052                                  WORKER::
000052                                          CALL    BRKCHK          ;CHECK TO SEE IF HE WANTS TO STOP
000056  032703  000001                          BIT     #T.OPFN,R3      ;OPERATOR?
000062  001076                                 BNE     EXPROP          ;YES, FINE
000064  032703  040000                          BIT     #T.OTHR,R3      ;CHARACTER?
000070  001002                                 BNE     1$              ;YES, WHAT IS IT?
000072  000175  177772                          JMP     @SI.RET(SI)     ;NO, RETURN
                                        ;+
                                        ; CHECK FOR SPECIAL CASES OF ASSIGNMENT AND RIGHT BRACKET
                                        ; (DIMENSION ARG FOR OP)
                                        ;-
000076  122704  000133                  1$:     CMPB    #CHRBKR,R4      ;BACK ARROW?
000102  001015                                 BNE     3$              ;IS NOT THEN SKIP
000104                                          CALL    GETNXT          ;GET NEXT TOKEN
000110  032703  001000                          BIT     #T.VAR,R3       ;VARIABLE?
000114  001002                                 BNE     2$              ;YES, FINE
000116  000137  000000G                        JMP     SPASSN          ;HANDLE SPECIAL ASSIGNMENTS
000122  010501                          2$:     MOV     SI,R1           ;POINT TO VAL
000124                                          CALL    ASSIGN          ;ASSIGN VAL TO VAR
000130                                          CALL    GETNXT          ;GET NEXT TOKEN
000134  000746                                 BR      WORKER          ;AND CYCLE

000136  122704  000047                  3$:     CMPB    #CHRRBR,R4      ;NO, RIGHT BRACKET?
000142  001402                                 BEQ     4$              ;YES, GET DIM ARG
000144  000175  177772                          JMP     @SI.RET(SI)     ;NO, JUST RETURN

000150                                  4$:     CALL    GETNXT          ;GET NEXT TOKEN
000154                                          CALL    EXPR            ;RECURSES! FOILED AGAIN!
000160  032765  100000  177774                  BIT     #T.VAL,SI.BIT(SI);VAL ON SI?
000166  001501                                 BEQ     EXPSYN          ;NO, SYNTAX ERROR
000170  016500  177776                          MOV     SI.VAL(SI),R0   ;POINT TO EXPRESSION
000174  132760  000006  000003                  BITB    #A.NUM,A.BITS(R0); NUMERIC DIMENSION ARG/FILE TYPE?
000202  001473                                 BEQ     EXPSYN          ; NOPE! SYNTAX ERROR
000204  022760  000001  000004                  CMP     #1,A.ALEN(R0)   ; SCALAR? (PERMIT FILE TYPE EXPRESSION!!)
000212  001067                                 BNE     EXPSYN          ;NO, DIM ARGS MUST BE SCALARS
000214  032703  040000                          BIT     #T.OTHR,R3      ;RETURN A CHAR?
000220  001464                                 BEQ     EXPSYN          ;NO, SYNTAX ERROR
000222  122704  000073                          CMPB    #CHRLBR,R4      ;YES, LEFT BRACKET?
000226  001061                                 BNE     EXPSYN          ;NO, SYNTAX ERROR
000230  012725  000372'                        MOV     #EXPSYN,(SI)+   ;PUSH ERROR RETURN
000234  010325                                 MOV     R3,(SI)+        ;PUSH BITS
000236  010425                                 MOV     R4,(SI)+        ;PUSH [ TO MARK DIM ARG
000240                                          CALL    GETNXT          ;GET NEXT TOKEN
000244  032703  000001                  EXPR1:  BIT     #T.OPFN,R3      ;OP OR FUNCTION?
000250  001450                                 BEQ     EXPSYN          ;NO, SYNTAX ERROR
000252  032703  000100                          BIT     #T.DIM,R3       ;OP TAKE DIMENSION ARG?
000256  001445                                 BEQ     EXPSYN          ;NO, SYNTAX ERROR
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 46
EXPR            EVALUTE APL EXPRESSION


                                        ;+
                                        ; AN OP HAS BEEN FOUND.  NOW CHECK IF IT SHOULD BE MONADIC
                                        ; OR DYADIC.
                                        ;-
000260  013700  003314'          EXPROP: MOV     SIFRAM,R0       ;POINT TO THIS FRAME
000264  052760  100000  000002          BIS     #F.PRNT,F.FLAG(R0);PRINT RESULT
000272  012725  000372'                  MOV     #EXPSYN,(SI)+   ;PUSH ERROR RETURN
000276  010325                          MOV     R3,(SI)+        ;PUSH BITS
000300  010425                          MOV     R4,(SI)+        ;PUSH OP NUMBER ON STACK
000302                                  CALL    GETNXT          ;GET NEXT TOKEN
000306  032703  105004                  BIT     #T.NIL+T.VAR+T.CON+T.VAL,R3;POSSIBLY A VAL?
000312  001021                          BNE     EXPDYD          ;YES, TRY FOR DYADIC OP
                                        ;+
                                        ; OP MUST BE MONADIC.  CHECK FOR SPECIAL CASES OF A DOT
                                        ; (INNER/OUTER PRODUCT), A SLASH (REDUCTION), OR A BACK SLASH (SCAN)
                                        ;-
000314  032703  040000                  BIT     #T.OTHR,R3      ;IS IT A CHARACTER?
000320  001413                          BEQ     1$              ;NOPE
000322  122704  000047                  CMPB    #CHRRBR,R4      ;]?
000326  001413                          BEQ     EXPDYD          ;YES, DYADIC OP! (WITH SUBSCRIPT ON LEFT!)
000330  022704  000042                  CMP     #CHRRPR,R4      ;)?
000334  001410                          BEQ     EXPDYD          ;YES, TRY FOR DYADIC OP?????
000336  022704  000056                  CMP     #CHRDOT,R4      ;YES, IS IT A DOT?
000342  001002                          BNE     1$              ;NO, Must be a Monadic Op
000344  000137  000000G                 JMP     DOT             ;YES, PROCESS INNER/OUTER PRODUCTS
                                        ;+
                                        ; PROCESS MONADIC OPERATOR
                                        ;-
000350                          1$:     CALL    MONAD           ;PROCESS OP
000354  000636                          BR      WORKER          ;AND CONTINUE
                                        ;+
                                        ; PROCESS DYADIC OP BUT WATCH FOR SPECIAL CASE MENTIONED
                                        ; BELOW
                                        ;-
000356                          EXPDYD: CALL    VAL             ;GET LEFT PART OF OP
000362  000404                          BR      EXPSPL          ;RATS! PROCESS SPECIAL CASE
000364                                  CALL    DYAD            ;PROCESS DYADIC OP
000370  000630                          BR      WORKER          ;AND CYCLE
```

```
                                ;+
                                ; THIS CODE PROCESSES THE UGLY CASE ENCOUNTERED WHEN PARSING
                                ; APL.  AN EXAMPLE IS ''-/[2]+A''.  AT THIS POINT WE HAVE SEEN THE
                                ; ''A'', THE ''+'', AND THE EXPRESSION IN BRACKETS.  UP TO NOW IT
                                ; LOOKS LIKE THE CASE  ''ARRAY[2]+A'' WHICH WOULD REQUIRE THE PLUS
                                ; OPERATOR TO BE DYADIC.  INSTEAD, WHEN WE FIND TH SLASH WHILE
                                ; PROCESSING THE LEFT OPERATOR, WE DISCOVER THAT THE PLUS SHOULD
                                ; BE MONADIC.  THE PROBLEM IS THAT WE HAVE ALREADY EVALUATED THE
                                ; EXPRESSION IN BRACKETS AND IT IS ON THE STACK WHERE THE RESULT
                                ; OF THE MONADIC RESULT OF ''+A'' SHOULD GO.  THE SOLUTION IS TO
                                ; POP THE DIMENSION ARG (EXPRESSION IN BRACKETS) FROM THE STACK
                                ; AND HIDE IT, DO THE MONADIC OPERATION, THEN PUT THE DIM ARG
                                ; BACK ON THE STACK.  WE CAN HIDE THE DIM ARG IN THE RESULT
                                ; FRAME FOR THIS EVALUATION FRAME.
                                ;               BLECH!
                                ;-
000372  104407                  EXPSYN: SYNERR                  ;SYNTAX ERROR CALLER

000374  013700  003314'         EXPSPL: MOV     SIFRAM,R0       ;POINT TO RESULT OF THIS FRAME
000400  014540                          MOV     -(SI),-(R0)     ;MOVE VALUE
000402  014540                          MOV     -(SI),-(R0)     ;MOVE BITS
000404  005045                          CLR     -(SI)           ;CLEAR RETURN
000406                                  CALL    MONAD           ;DO MONADIC OP
000412  013700  003314'                 MOV     SIFRAM,R0       ;POINT TO HIDDEN ARG
000416  062700  177774                  ADD     #SI.BIT,R0      ;POINT TO BITS
000422  012725  000372'                 MOV     #EXPSYN,(SI)+   ;PUSH ERROR RETURN
000426  011025                          MOV     @R0,(SI)+       ;MOVE BITS
000430  005020                          CLR     (R0)+           ;CLEAR HIDING PLACE
000432  011025                          MOV     @R0,(SI)+       ;MOVE VALUE BITS
000434  005010                          CLR     @R0             ;AND CLEAR HIDING PLACE
000436  012725  000372'                 MOV     #EXPSYN,(SI)+   ;PUSH ERROR RETURN
000442  012725  040000                  MOV     #T.OTHR,(SI)+   ;PUSH CHARACTER INDICATOR
000446  012725  000073                  MOV     #CHRLBR,(SI)+   ;PUSH [ TO MARK DIMENSION ARG
000452  000674                          BR      EXPR1           ;AND PROCESS DIM ARG FOR OP
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 49
EXPR            EVALUTE APL EXPRESSION

     1          000001                          .END
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 49-1
SYMBOL TABLE

AC0    =%000000          CHRFIL= 000177    C.ALOC= 000162    D.CONT= 000006    EDICNT  005064R  025
AC1    =%000001          CHRFRO= 000135    C.BLOK= 000064    D.CTLM= 000000    EDIFLG  005062R  025
AC2    =%000002          CHRLBR= 000073    C.BLSZ= 000140    D.DEAL= 000003    EDIT   = 000003
AC3    =%000003          CHRLPR= 000072    C.BUFF= 177760    D.DECD= 000002    EIS    = 000000
AC4    =%000004          CHRMNS= 000137    C.BUFL= 177762    D.DIV = 000000    ELESIZ  005132R  025
AC5    =%000005          CHRMUL= 000075    C.CCNT= 177766    D.DMD = 000010    ENDCOR  000004R  025
ALLINT  003274R  025    CHRNEG= 000100    C.CPTR= 177764    D.DQAD= 177777    EOF    = 104475
ARYRET  003276R  025    CHRPDL= 000271    C.CZ   = 001000   D.DROP= 000002    ERRMAX= 000166
ASCQUO= 000042          CHRPLU= 000055    C.DIG  = 000040   D.ENCD= 000002    ERRNUM  004772R  025
ASSIGN= ****** GX       CHRPWR= 000120    C.FASW= 020000    D.EQ   = 000000    ERRPTR  004770R  025
A.ALEN= 000004          CHRQST= 000121    C.FLEN= 000010    D.EXPD= 000002    ERR0    004774R  025
A.ARG  = 000040         CHRQUO= 000113    C.FLSH= 004000    D.FGET= 000005    EXEBUF  005054R  025
A.ASB  = 000200         CHRRBR= 000047    C.FRC  = 000001   D.FMT  = 000004    EXPDYD  000356R  020
A.BITS= 000003          CHRRPR= 000042    C.IDEN= 000010    D.FPNT= 000005    EXPR    000000RG  020
A.CON  = 000020         CHRSEM= 000074    C.NAME= 000102    D.FPUT= 000005    EXPROP  000260R  020
A.DIMS= 000002          CHRSMO= 000112    C.NOBF= 000400    D.GE   = 000000   EXPR1   000244R  020
A.DIM1= 000006          CHRSPA= 000040    C.NUM  = 000020   D.GT   = 000000   EXPSPL  000374R  020
A.NUM  = 000006         CHRUPA= 000131    C.OP   = 000004   D.IBM  = 000002   EXPSYN  000372R  020
A.PTR  = 000000         CHR.BS= 000010    C.OPN  = 010000   D.IND  = 000002   E$CT    005056R  025
A.TMP  = 000010         CHR.CC= 000003    C.RDLN= 177776    D.INNR= 000013    E$PT    005060R  025
BADDIR= 104463          CHR.CR= 000015    C.REWT= 002000    D.INTR= 000006    FA.APD= 000100
BADERR= 104437          CHR.CZ= 000032    C.RND  = 000010   D.LE   = 000000   FA.CRE= 000010
BLZBUB  005116R  025    CHR.GT= 000046    C.SBIT= 177756    D.LOG  = 000001   FA.DLK= 001000
BRKCHK= ****** GX       CHR.LF= 000012    C.SPFX= 000100    D.LT   = 000000   FA.ENB= 100000
BRSELF  005046R  025    CHR.LT= 000043    C.SPGL= 000200    D.MAX  = 000000   FA.EXC= 002000
CCNT    005072R  025    CHR.SI= 000017    C.SPLX= 000001    D.MEM  = 000002   FA.EXT= 000004
CHANEX  003060R  025    CHR.SL= 000057    C.SPRS= 000002    D.MIN  = 000000   FA.NSP= 000100
CHAN00  000256R  025    CHR.SO= 000016    C.SRC  = 177774   D.MUL  = 000000   FA.NSR= 000200
CHAN01  000330R  025    CHR.TB= 000011    C.SRCP= 000024    D.NE   = 000000   FA.POS= 010000
CHAN02  000512R  025    CHR.UL= 000106    C.TT   = 000004   D.NN   = 000000   FA.RD  = 000001
CHAN03  000674R  025    CHR..A= 000141    C.WIDC= 177770    D.NR   = 000000   FA.RWD= 004000
CHAN04  001056R  025    CHR..E= 000145    C.WIDR= 177772    D.OR   = 000000   FA.SEQ= 040000
CHAN05  001240R  025    CHR..S= 000163    DEBUG  = 000000   D.PLUS= 000000    FA.SHR= 000040
CHAN06  001422R  025    CHR..T= 000164    DEFERR= 104405    D.PSUB= 000006    FA.TMP= 000020
CHAN07  001604R  025    CHR..U= 000165    DEVIL   005114R  025 D.PWR  = 000001    FA.WCK= 020000
CHAN08  001766R  025    CHR..Z= 000172    DEVNTF= 104470    D.QQAD= 000000    FA.WRT= 000002
CHAN09  002150R  025    CHR..0= 000060    DIGITS  000006R  025 D.QUAD= 000000    FD.BLK= 000010
CHAN10  002332R  025    CHR..1= 000061    DIMAX  = 000020   D.RES  = 000000    FD.CCL= 000002
CHAN11  002514R  025    CH.AND= 000001    DIMBLK  003112R  025 D.RHO  = 000000    FD.COM= 020000
CHAN12  002676R  025    CMDBUF  003250R  025 DIMPTR  003114R  025 D.ROT  = 000003    FD.CR  = 000002
CHAN13  000022R  023    CMDMAX= 000010    DIMTYP  003116R  025 D.SCON= 000006    FD.DIR= 000010
CHAN14  000204R  023    CNTSIZ  005130R  025 DIRFUL= 104466    D.SDIF= 000006    FD.FTN= 000001
CHNOPN= 104471          COMBIN  005076R  025 DOMERR= 104417    D.SUB  = 000000   FD.F11= 040000
CHRAND= 000051          COMCNT  005126R  025 DOT    = ****** GX   D.SUBS= 000006    FD.INS= 000010
CHRBEL= 000007          COMERR= 104425    DPTERR= 104416    D.TAKE= 000002    FD.ISP= 002000
CHRBKR= 000133          COMPTR  005122R  025 DYAD   = ****** GX   D.TRAN= 000004    FD.MNT= 100000
CHRBOL= 000200          COMRTN  003156R  025 DYDCNT  003224R  025 D.UNON= 000006    FD.OSP= 004000
CHRBOX= 000114          COMSKP  005124R  025 D.AND  = 000000    D.$MAX= 000000    FD.PLC= 000004
CHRCMA= 000054          CORPTR  005120R  025 D.BXCT= 000007    D.$MIN= 000000    FD.PRN= 000004
CHRCOL= 000076          CS.ASS= 000200    D.BXIO= 000007    D.$MUL= 000000    FD.PSE= 010000
CHRDEL= 000107          CS.BOO= 000010    D.BXNL= 000007    D.$PLU= 000000    FD.RAH= 000001
CHRDIV= 000053          CS.FLO= 000020    D.BX  = 000007    D.$RES= 000000    FD.RAN= 000002
CHRDLR= 000176          CS.IDN= 000040    D.BXHW= 000007    D.$SUB= 000000    FD.REC= 000001
CHRDLT= 000110          CS.ONE= 000100    D.BXRL= 000007    D.1COM= 000001    FD.RWM= 000001
CHRDOT= 000056          CTEPS   000152R  025 D.CIRC= 000001    D.1EXP= 000002    FD.SDI= 000020
CHREP  = 000105         CTMAX   000212R  025 D.COMB= 000001    D.1ROT= 000003    FD.SQD= 000040
CHRERR= 104431          CTRMAX  000172R  025 D.COMP= 000001    EDICHR  005063R  025 FD.TTY= 000004
```

SYMBOL TABLE

```
FD.WBH= 000002          F.EFN = 000050          F.VAL = 177776          L.PRNT= 000001          M.PITM= 000000
FF.CHR= 000005          F.END = 002000          F.VBN = 000064          L.PTR = 000000          M.PRDS= 000006
FF.NV = 000003          F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004          M.QQAD= 000000
FF.POE= 000002          F.ERR = 000052          GETNXT= ****** GX       L.SUBS= 000002          M.QUAD= 000000
FF.RWD= 000001          F.FACC= 000043          GNOME   003102R 025     MACHIN= 000106          M.RAND= 000012
FF.RWF= 000006          F.FFBY= 000014          HASHC   005136R 025     MAXCHN= 000014          M.RAVL= 000000
FF.SPC= 000004          F.FLAG= 000002          HASHTB  005140R 025     MONAD = ****** GX       M.RCIP= 000000
FHEADR  005012R 025     F.FLB = 000100          HSHMSK= 177601          MONBLK  003120R 025     M.RDCT= 000011
FHRET   005040R 025     F.FNAM= 000110          HSHSIZ= 000200          MONCVT  003132R 025     M.REVS= 000003
FILNTF= 104467          F.FNB = 000102          ICHAN   005050R 025     MONLEN  003130R 025     M.ROLL= 000000
FIS   = 000000          F.FNDF= 000010          ILFNAM= 104464          MONPTR  003122R 025     M.SATA= 000001
FISCON= 046000          F.FTYP= 000116          ILLCHN= 104540          MONSFT  003126R 025     M.SCAN= 000011
FIXPTR  003074R 025     F.FUN = 000010          ILLSW = 104565          MONTYP  003124R 025     M.SHAP= 000000
FLAGS   004764R 025     F.FVER= 000120          INDERR= 104410          MONVAL  003134R 025     M.SIGN= 000000
FLCBUF  003250R 025     F.HEAD= 000006          INNTMP  003250R 025     M.ABS = 000000          M.SUBR= 000001
FLINE   005014R 025     F.HIBK= 000004          INNTM2  005016R 025     M.BOL = 000200          M.SYM = 000203
FLOEXP  003270R 025     F.LBL = 000020          INPTR   000064R 025     M.BXAV= 000007          M.SYN = 000201
FLONEG  003272R 025     F.LINE= 000012          INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FLSWTB  000222R 025     F.LOC = 000004          IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FNINSI  005044R 025     F.LUN = 000042          I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.RD = 000001          F.MON = 000002          I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FPMODE  000042R 025     F.NREC= 000030          LABLPT  005042R 025     M.BXRL= 000007          NAMBUF  003240R 025
FPP   = 000000          F.OPN = 000010          LARGBL  003162R 025     M.BXTS= 000007          NB.DEV= 000200
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R 025     M.BXWA= 000007          NB.DIR= 000100
FREBLK  007400R 025     F.PREV= 000000          LARGFL  003160R 025     M.CEI = 177777          NB.NAM= 000004
FUZZ    000142R 025     F.PRNT= 100000          LARGLE  003174R 025     M.CHR = 000202          NB.SD1= 000400
FXPTR1  003076R 025     F.PROT= 010000          LARGLN  003166R 025     M.CMT = 000206          NB.SD2= 001000
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R 025     M.DOUT= 000006          NB.SNM= 000040
F$EVI = 000200          F.RATT= 000001          LARGSF  003172R 025     M.DQAD= 000006          NB.STP= 000020
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R 025     M.ENCD= 000004          NB.SVR= 000010
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R 025     M.EXE = 000000          NB.TYP= 000002
F.ACTL= 000076          F.READ= 004000          LBINC   005100R 025     M.EXP = 000001          NB.VER= 000001
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R 025     M.FACT= 000001          ND.DEV= 000200
F.BBFS= 000062          F.RES = 000040          LBUFLN= 000621          M.FIL = 000177          ND.EXT= 000002
F.BDB = 000070          F.RET = 177772          LCOUNT  005106R 025     M.FLR = 177777          ND.NAM= 000004
F.BGBC= 000057          F.RSIZ= 000002          LENERR= 104412          M.FMT = 000004          ND.PPN= 000100
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R 025     M.GDUP= 000003          ND.$EX= 000000
F.BKDN= 000026          F.R2  = 000014          LEXPTR  003100R 025     M.GDWN= 000003          ND.$E$= 000020
F.BKDS= 000020          F.R3  = 000016          LFTOP   005066R 025     M.GETI= 000006          ND.$NM= 000000
F.BKEF= 000050          F.R4  = 000020          LINC    005102R 025     M.IBM = 000003          ND.$N$= 000040
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R 025     M.IDN = 000000          ND.$PN= 001400
F.BKST= 000024          F.SPDV= 000072          LININC  005026R 025     M.INSP= 000006          NEGSGN  005063R 025
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020          M.IOTA= 000000          NINPT1  003302R 025
F.BR  = 000020          F.STBK= 000036          LITOP   005072R 025     M.LIT = 000205          NINPT2  003300R 025
F.CHR = 000075          F.STK = 000022          LNLNG = 104541          M.LN  = 000001          NINSTK  003304R 025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R 025     M.MMD = 000010          NINTMP  003310R 025
F.CODE= 000010          F.STP = 000001          LOWCOR  000002R 025     M.NEG = 000000          NINTYP  003306R 025
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R 025     M.NOT = 000000          NOTAVL= 104472
F.DSPT= 000044          F.TRC = 000002          LWIDTH  000070R 025     M.NUB = 000006          NOTOPN= 104473
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003          M.NUM = 000204          NPRT1   004762R 025
F.EFBK= 000010          F.URBD= 000020          L.CNT = 000002          M.OUTR= 000012          NULL    007364R 025
```

SYMBOL TABLE

```
NUMCHN= 000020          ORIGIN  000030R     025 R.FIX  = 000001          SYSTAT  000000R     024 T.LIST= 000400
NUMDFL  003230R     025 OUTPTR  000066R     025 R.LCT  = 000002          SY.BAD= 000001          T.LIT = 000020
NUMEFL  003233R     025 OVROPR= 000000          R.LN   = 000002          SY.BRK= 100000          T.MON = 000010
NUMESN  003236R     025 PATCH   007344R     025 R.LNUM= 000010          SY.BS = 000010          T.NIL = 000004
NUMFLG  003230R     025 PRECSN  000052R     025 R.NEXT= 000006          SY.CLR= 000040          T.NUM = 002000
NUMLEN  000062R     025 RANDAC  003064R     025 R.OFF  = 000000          SY.ERP= 000020          T.OPFN= 000001
NUMNEG  003232R     025 RANDY   003062R     025 R.PREV= 000004          SY.NOU= 000100          T.OTHR= 040000
NUMQFL  003234R     025 RANERR= 104411          R.PTR  = 000000          SY.OPN= 000002          T.SOP = 000002
NUMRIG  003231R     025 RARGBL  003204R     025 R.SEQ  = 000003          SY.PMT= 001000          T.SPOP= 000040
NUMWID  003226R     025 RARGCV  003220R     025 R.VAR  = 000002          SY.SRT= 000400          T.TMP = 000200
NUMZFL  003235R     025 RARGFL  003202R     025 R6     =%000006          SY.SVC= 000004          T.VAL = 100000
NYIERR= 104432          RARGLE  003216R     025 R7     =%000007          SY.SVL= 000200          T.VAR = 001000
N.BLSZ= 000036          RARGLN  003210R     025 SI     =%000005          S.AB   = 177776          UPPER = 177400
N.DESC= 000020          RARGPT  003206R     025 SIDAMN= 104433           S.ASB  = 100000          UTNAME  000344R     023
N.DEV = 000032          RARGSF  003214R     025 SIFRAM  003314R     025 S.BFHD= 000020          VAL    = ****** GX
N.DEVN= 000034          RARGTY  003212R     025 SISTAK  006350R     025 S.BITS= 000002          VALERR= 104413
N.DID = 000024          RARGVA  003222R     025 SITOP   006346R     025 S.CHR1= 000003          VARBLE  007400R     025
N.DVNM= 000032          RBASE   005074R     025 SITP    007144R     025 S.FATT= 000016          VARBLS  000000R     025
N.EXT = 000014          RCOUNR  005112R     025 SIWARN= 000200           S.FDB  = 000140          VERNUM  000000R     025
N.FID = 000000          RCOUNT  005110R     025 SIZTAB  000044R     025 S.FLB  = 040000          VRSION= 000002
N.FNAM= 000006          REDCNT  005070R     025 SI.BIT= 177774           S.FNAM= 000006          WIDTH   000072R     025
N.FTYP= 000014          REDIGN  005066R     025 SI.LEN= 177772           S.FNB  = 000036          WORKER  000052RG    020
N.FVER= 000016          REDOFF  005072R     025 SI.RET= 177772           S.FNBW= 000017          WROUND  000020R     025
N.NAME= 000006          REENTR  003060R     025 SI.VAL= 177776           S.FNTY= 000004          WSDATE  000136R     025
N.NEXT= 000022          RESBIT  003142R     025 SKPSIZ  005134R     025 S.FTYP= 000002          WSFULL= 104403
N.PPN = 000024          RESCVT  003152R     025 SPASSN= ****** GX        S.LEN  = 000002          WSNAME  000074R     025
N.STAT= 000020          RESERR= 104440          SPHOLD  000004R     024 S.LINK= 177774          WSTIME  000132R     025
N.UNIT= 000034          RESLEN  003150R     025 SPINIT  000002R     024 S.NFEN= 000020          $FIS   = ****** G
OBUFLN= 001000          RESPTR  003140R     025 STKINC= 000774           S.TYPE= 140000          $FPP   = ****** G
OCHAN   005052R     025 RESSFT  003146R     025 STOP   = 104420         S.VAL  = 000000          $FPPLN= ****** G
OMODE   000402R     023 RESTYP  003144R     025 SUBCNT  003104R     025 TKDIMS  005072R     025 $MACHN= ****** G
OPTMP0  005066R     025 RESULT  003136R     025 SUBDCT  003106R     025 TKFRAM  005066R     025 $RSTS  = ****** G
OPTMP1  005070R     025 RESVAL  003154R     025 SUBPTR  003110R     025 TKLEFT  005070R     025 $RSX   = ****** G
OPTMP2  005072R     025 RINC    005104R     025 SYMB1   005344R     025 TMODE   000403R     023 $RT11  = ****** G
OPTMP3  005074R     025 RLTBAS  005066R     025 SYMF    005340R     025 TMPTAG= 000166          .BOOLE= 000002
OPTMP4  005076R     025 ROTINC  005070R     025 SYMPT1  005346R     025 TTYBUF  004140R     025 .CHARA= 000000
OPTMP5  005100R     025 ROUND   000010R     025 SYMTAB  005352R     025 TTYLEN= 000622          .DEF.  = 000002
OPTMP6  005102R     025 ROVER   003312R     025 SYMTBF= 104404           TURKEY= 104436          .FLOAT= 000006
OPTMP7  005104R     025 RSTS   = 000000          SYMTOP  005350R     025 TYPE   = 000166          .FSRCB= ****** G
OPTMP8  005106R     025 RSX    = 000115          SYMTP   006272R     025 T.BOL  = 020000          .INTEG= 000004
OPTMP9  005110R     025 RTOP    005070R     025 SYMWRN= 000054           T.CON  = 004000          .MOLUN  000404R     023
OPTM10  005112R     025 RT11   = 000000          SYNERR= 104407           T.DIM  = 000100          ...GBL= 000000
OPTM14  005026R     025 R.BITS= 000003          SYSERR= 104402           T.FUN  = 010000          ...TPC= 000020

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
```

```
EXPR    MACRO M1113  08-FEB-80 12:15  PAGE 49-4
SYMBOL TABLE

TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000454      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12989 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:34
EXPR,DM1:[305,301]EXPR/-SP=APLRX1,APLDEF,EXPR,END
```

```
**FILE**ID**FIXLIN
```

```
FFFFFFFFFF      IIIIII      XX      XX    LL              IIIIII      NN      NN
FFFFFFFFFF      IIIIII      XX      XX    LL              IIIIII      NN      NN
FF                II        XX      XX    LL                II        NN      NN
FF                II        XX      XX    LL                II        NN      NN
FF                II          XX  XX      LL                II        NNNN    NN
FF                II          XX  XX      LL                II        NNNN    NN
FFFFFFFF          II            XX        LL                II        NN  NN  NN
FFFFFFFF          II            XX        LL                II        NN  NN  NN
FF                II          XX  XX      LL                II        NN    NNNN
FF                II          XX  XX      LL                II        NN    NNNN
FF                II        XX      XX    LL                II        NN      NN      ....
FF                II        XX      XX    LL                II        NN      NN      ....
FF              IIIIII      XX      XX    LLLLLLLLLL      IIIIII      NN      NN      ....
FF              IIIIII      XX      XX    LLLLLLLLLL      IIIIII      NN      NN      ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 12:18
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF  THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS     OPERATOR OVERLAYS
     5-  268    DEFINITIONS     REGISTERS
     6-  331    DEFINITIONS     MEMORY ORGANIZATION
     7-  377    DEFINITIONS     ORDERING PSECTS
     8-  426    DEFINITIONS     SI PARSE FRAMES
     9-  474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS     SI EXECUTION FRAMES
    11-  580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS     CHRTAB BITS
    12-  624    DEFINITIONS     SYSTEM STATUS BITS
    13-  640    DEFINITIONS     F-FRAMES
    14-  724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS     LIST STORAGE BLOCK
    18-  911    DEFINITIONS     INDEXING STACK FRAME
    19-  940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS     CHANNEL BLOCK
    22- 1085    DEFINITIONS             RSX DATA
    23- 1110    DEFINITIONS     NAME BLOCKS
    33- 1612    DEFINITIONS     MACROS
    35- 1905    DEFINITIONS     APL CHARACTERS
    36- 1967    DEFINITIONS     ERRORS
    44-    4    FIXLIN          FIXUP INPUT LINE.
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     =  'M         ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
  2513                                  .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
                        .TITLE  FIXLIN
                        .IDENT  \V02.03\

                        .SBTTL  FIXLIN          FIXUP INPUT LINE.

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;       First pass of line interpretation. This routine figures out
                        ;       what the line the guy typed looks like on his terminal.
                        ;       Spacing and backspacing is resolved, mnemonics are translated
                        ;       overstrikes are reduced to one byte. Backspaces make the final
                        ;       line shorter, and it can never get longer, so the "fixing" can
                        ;       be done in place in the line buffer.  First backspaces are
                        ;       rsolved, and then if the current terminal is in TTY mode, the
                        ;       mnemonics are translated.
                        ;
                        ; CALLING SEQUENCE:
                        ;       MOV     #<Line buffer>,R0
                        ;       CALL    FIXLIN
                        ;
                        ; INPUTS:
                        ;       R0      - Contains a pointer to the line buffer
                        ;
                        ; IMPLICIT INPUTS:
                        ;       TMODE   - Indicates the current terminal mode
                        ;
                        ; OUTPUTS:
                        ;       R0      - Contains a pointer to the line buffer
                        ;       R1      - Contains the byte count
                        ;       R4      - Contains a pointer past the last character
                        ;                 in the buffer.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;       NONE
                        ;
                        ; ROUTINES CALLED:
                        ;       BINSCH  - used to search overstrike/mneumonic tables
                        ;       CLRSI   - used in case an O<backspace>U<backspace>T is found
                        ;
                        ; SIDE EFFECTS:
                        ;       NONE
                        ;
                        ; REWRITTEN BY: Peter Haynes             DATE: May 23, 1979
                        ;--
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 45
FIXLIN          FIXUP INPUT LINE.

     46 000000                          .PSECT CODE
     47
     48 000000                          FIXLIN::
     49 000000  113746  000403'                MOVB    TMODE,-(SP)     ;Save the current terminal mode
     50 000004  010037  003074'                MOV     R0,FIXPTR       ;Save the pointer to the left margin
     51 000010  010003                         MOV     R0,R3           ;Initialize the Input pointer
     52 000012  010004                         MOV     R0,R4           ;                Output pointer
     53 000014  005037  003076'                CLR     FXPTR1          ;                Right margin pointer
     54
     55 000020  111302                  FIX1:  MOVB    (R3),R2         ;Get a character and
     56 000022  112723  000040                 MOVB    #CHRSPA,(R3)+   ;Replace it with a space
     57 000026  042702  177400                 BIC     #UPPER,R2       ;Clear out any high bits
     58 000032  120227  000015                 CMPB    R2,#CHR.CR      ;Is it a carriage return?
     59 000036  001472                         BEQ     8$              ;Yes
     60 000040  120227  000010                 CMPB    R2,#CHR.BS      ;Is it a backspace?
     61 000044  001012                         BNE     2$              ;No
     62 000046  020437  003074'                CMP     R4,FIXPTR       ;Yes, Are we at the left margin?
     63 000052  001762                         BEQ     FIX1            ;Yes, Then ignore it
     64 000054  020437  003076'                CMP     R4,FXPTR1       ;Are we at the right margin?
     65 000060  101402                         BLOS    1$              ;No
     66 000062  010437  003076'                MOV     R4,FXPTR1       ;Yes, Save pointer to right margin
     67 000066  005304                  1$:    DEC     R4              ;Backup output pointer
     68 000070  000753                         BR      FIX1            ;Loop for next character
     69                                 ;
     70                                 ;  Process a backspace
     71                                 ;
     72 000072  122702  000040          2$:    CMPB    #CHRSPA,R2      ;Is the character a space?
     73 000076  001402                         BEQ     3$              ;Yes
     74 000100  121402                         CMPB    (R4),R2         ;Does the character = the output characcter?
     75 000102  001002                         BNE     4$              ;No
     76 000104  005204                  3$:    INC     R4              ;Increment output pointer
     77 000106  000744                         BR      FIX1            ;And loop
     78
     79 000110  121427  000040          4$:    CMPB    (R4),#CHRSPA    ;Is the output character a space?
     80 000114  001441                         BEQ     7$              ;Yes
     81                                 ;
     82                                 ;  Check for a legal overstrike
     83                                 ;
     84 000116  105737  000403'          5$:   TSTB    TMODE           ;Are we in APL mode?
     85 000122  001147                         BNE     FIXERR          ;No, Can't have overstrikes in TTY mode
     86 000124  010200                         MOV     R2,R0           ;Copy the input character
     87 000126                                 ASH     #8.,R0          ;Move it to the upper byte
     88 000142  151400                         BISB    (R4),R0         ;Combine it with the output character
     89 000144  120214                         CMPB    R2,(R4)         ;Which is bigger?
     90 000146  103401                         BLO     6$              ;The output character
     91 000150  000300                         SWAB    R0              ;The input character, switch them around
     92 000152                          6$:    PUSH    R3              ;Save the input pointer
     93 000154  012701  000000G                MOV     #OVRTAB,R1      ;Setup the search
     94 000160  012703  000000G                MOV     #OVRTND,R3      ;parameters
     95 000164                                 CALL    BINSCH          ;Search for the overstrike, Is it legal?
     96 000170  001124                         BNE     FIXERR          ;No, Character error
     97 000172                                 POP     R3              ;Restore the input pointer
     98 000174  062702  000000G                ADD     #OVSTRK,R2      ;Convert the character to an overstrike
     99 000200  120227  000000G                CMPB    R2,#CHREND      ;Is it out of range?
    100 000204  103116                         BHIS    FIXERR          ;Yes, Character error
    101 000206  120227  000000G                CMPB    R2,#CHR.OU      ;Is it the escape character?
    102 000212  001002                         BNE     7$              ;No
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 45-1
FIXLIN          FIXUP INPUT LINE.

    103 000214  000137  000000G                 JMP     CLRSI           ;Yes, So Escape
    104
    105 000220  110224                  7$:     MOVB    R2,(R4)+        ;Store it away
    106 000222  000676                          BR      FIX1            ;And loop
```

FIXLIN   FIXUP INPUT LINE.

```
                                        ;
                                        ;  Process a carriage return
                                        ;
000224  020437  003076'         8$:     CMP     R4,FXPTR1       ;Are we inside the line?
000230  103002                          BHIS    FIXEND          ;No, Ok
000232  013704  003076'                 MOV     FXPTR1,R4       ;Yes, Must have backspace into it
000236  110224                  FIXEND: MOVB    R2,(R4)+        ;Store the carriage return
000240  013700  003074'                 MOV     FIXPTR,R0       ;Point to the left margin
000244  010401                          MOV     R4,R1           ;Copy the output pointer
000246  160001                          SUB     R0,R1           ;Get t!  number of characters
000250  105726                          TSTB    (SP)+           ;Are we in TTY terminal mode?
000252  001001                          BNE     1$              ;Yes, Then do pass two
000254                                  RETURN                  ;No, we're done
                                        ;
                                        ;  Pass two, Translate TTY mnemonics
                                        ;
000256  013703  003074'         1$:     MOV     FIXPTR,R3       ;Point to left margin
000262  010304                          MOV     R3,R4           ;Output pointer
000264  005046                          CLR     -(SP)           ;Fake terminal mode to APL

000266  112300                  FIX2:   MOVB    (R3)+,R0        ;Get a character
000270                                  CALL    FIXSHF          ;Convert lower case to upper case letters
000274  010002                          MOV     R0,R2           ;Copy the character
000276  120227  000056                  CMPB    R2,#CHRDOT      ;Is it a dot?
000302  001044                          BNE     2$              ;No
000304                                  PUSH    R3              ;Yes, A mnemonic maybe?, Save the input pointer
000306  112300                          MOVB    (R3)+,R0        ;Get the first character
000310  120027  000015                  CMPB    R0,#CHR.CR      ;Is it a carriage return?
000314  001434                          BEQ     1$              ;Yes
000316                                  CALL    FIXSHF          ;No, Convert lower to upper case
000322                                  ASH     #8.,R0          ;Move it to the upper byte
000336  121327  000015                  CMPB    (R3),#CHR.CR    ;Is the second character a carriage return?
000342  001421                          BEQ     1$              ;Yes
000344  152300                          BISB    (R3)+,R0        ;No, Combine it with the first character
000346                                  CALL    FIXSHF          ;Convert it to upper case
000352  000300                          SWAB    R0              ;Reverse their order
000354  012701  000000G               MOV     #MNTAB,R1       ;Setup the search
000360  012703  000000G               MOV     #MNTBND,R3      ;Parameters
000364                                  CALL    BINSCH          ;Search for mnemonic, Is it legal?
000370  001006                          BNE     1$              ;No, Then dot must be a dot only
000372  116202  000000G               MOVB    MNTRTB(R2),R2   ;Translate mnemonic to APL character
000376                                  POP     R3              ;Restore input pointer
000400  062703  000002                  ADD     #2,R3           ;Increment input pointer
000404  000407                          BR      3$              ;Go to finish up

000406                          1$:     POP     R3              ;Get back old input pointer
000410  012702  000056                  MOV     #CHRDOT,R2      ;Get dot back too
000414  042702  177400          2$:     BIC     #UPPER,R2       ;Clean out garbage bits
000420  116202  000000G                 MOVB    TRNTAB(R2),R2   ;Translate ASCII to APL
000424  042702  177400          3$:     BIC     #UPPER,R2       ;Clean out garbage again
000430  020227  000015                  CMP     R2,#CHR.CR      ;Is the character a carriage return?
000434  001700                          BEQ     FIXEND          ;Yes, Then go to return
000436  110224                          MOVB    R2,(R4)+        ;No, Store the character in the output line
000440  000712                          BR      FIX2            ;Loop for more characters
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 47
FIXLIN          FIXUP INPUT LINE.

    163 000442  104431                FIXERR: CHRERR                  ;Character Error
    164
    165                                 ;
    166                                 ;  This routine converts the character in R0 to upper case if it is a letter
    167                                 ;
    168 000444  120027  000141          FIXSHF: CMPB    R0,#CHR..A      ;Is it an "a" or bigger?
    169 000450  002405                          BLT     1$              ;No
    170 000452  120027  000172                  CMPB    R0,#CHR..Z      ;Is it a "z" or smaller?
    171 000456  003002                          BGT     1$              ;No
    172 000460  142700  000040                  BICB    #CHRSPA,R0      ;It's a letter, Convert it to upper case
    173 000464                          1$:     RETURN
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 48
FIXLIN           FIXUP INPUT LINE.

      1          000001                              .END
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 48-1
SYMBOL TABLE

AC0    =%000000          CHRFIL= 000177          CTMAX   000212R  025 D.CONT= 000006          EDICNT  005064R  025
AC1    =%000001          CHRFRO= 000135          CTRMAX  000172R  025 D.CTLM= 000000          EDIFLG  005062R  025
AC2    =%000002          CHRLBR= 000073              C.ALOC= 000162      D.DEAL= 000003          EDIT   = 000003
AC3    =%000003          CHRLPR= 000072              C.BLOK= 000064      D.DECD= 000002          EIS    = 000000
AC4    =%000004          CHRMNS= 000137              C.BLSZ= 000140      D.DIV = 000000          ELESIZ  005132R  025
AC5    =%000005          CHRMUL= 000075              C.BUFF= 177760      D.DMD = 000010          ENDCOR  000004R  025
ALLINT  003274R  025 CHRNEG= 000100              C.BUFL= 177762      D.DQAD= 177777          EOF    = 104475
ARYRET  003276R  025 CHRPDL= 000271              C.CCNT= 177766      D.DROP= 000002          ERRMAX= 000166
ASCQUO= 000042          CHRPLU= 000055              C.CPTR= 177764      D.ENCD= 000002          ERRNUM  004772R  025
A.ALEN= 000004          CHRPWR= 000120              C.CZ  = 001000      D.EQ  = 000000          ERRPTR  004770R  025
A.ARG = 000040          CHRQST= 000121              C.DIG = 000040      D.EXPD= 000002          ERRO    004774R  025
A.ASB = 000200          CHRQUO= 000113              C.FASW= 020000      D.FGET= 000005          EXEBUF  005054R  025
A.BITS= 000003          CHRRBR= 000047              C.FLEN= 000010      D.FMT = 000004          E$CT    005056R  025
A.CON = 000020          CHRRPR= 000042              C.FLSH= 004000      D.FPNT= 000005          E$PT    005060R  025
A.DIMS= 000002          CHRSEM= 000074              C.FRC = 000001      D.FPUT= 000005          FA.APD= 000100
A.DIM1= 000006          CHRSMO= 000112              C.IDEN= 000010      D.GE  = 000000          FA.CRE= 000010
A.NUM = 000006          CHRSPA= 000040              C.NAME= 000102      D.GT  = 000000          FA.DLK= 001000
A.PTR = 000000          CHRUPA= 000131              C.NOBF= 000400      D.IBM = 000002          FA.ENB= 100000
A.TMP = 000010          CHR.BS= 000010              C.NUM = 000020      D.IND = 000002          FA.EXC= 002000
BADDIR= 104463          CHR.CC= 000003              C.OP  = 000004      D.INNR= 000013          FA.EXT= 000004
BADERR= 104437          CHR.CR= 000015              C.OPN = 010000      D.INTR= 000006          FA.NSP= 000100
BINSCH= ****** GX       CHR.CZ= 000032              C.RDLN= 177776      D.LE  = 000000          FA.NSR= 000200
BLZBUB  005116R  025 CHR.GT= 000046              C.REWT= 002000      D.LOG = 000001          FA.POS= 010000
BRSELF  005046R  025 CHR.LF= 000012              C.RND = 000010      D.LT  = 000000          FA.RD = 000001
CCNT    005072R  025 CHR.LT= 000043              C.SBIT= 177756      D.MAX = 000000          FA.RWD= 004000
CHANEX  003060R  025 CHR.OU= ****** GX           C.SPFX= 000100      D.MEM = 000002          FA.SEQ= 040000
CHAN00  000256R  025 CHR.SI= 000017              C.SPGL= 000200      D.MIN = 000000          FA.SHR= 000040
CHAN01  000330R  025 CHR.SL= 000057              C.SPLX= 000001      D.MUL = 000000          FA.TMP= 000020
CHAN02  000512R  025 CHR.SO= 000016              C.SPRS= 000002      D.NE  = 000000          FA.WCK= 020000
CHAN03  000674R  025 CHR.TB= 000011              C.SRC = 177774      D.NN  = 000000          FA.WRT= 000002
CHAN04  001056R  025 CHR.UL= 000106              C.SRCP= 000024      D.NR  = 000000          FD.BLK= 000010
CHAN05  001240R  025 CHR..A= 000141              C.TT  = 000004      D.OR  = 000000          FD.CCL= 000002
CHAN06  001422R  025 CHR..E= 000145              C.WIDC= 177770      D.PLUS= 000000          FD.COM= 020000
CHAN07  001604R  025 CHR..S= 000163              C.WIDR= 177772      D.PSUB= 000006          FD.CR = 000002
CHAN08  001766R  025 CHR..T= 000164              DEBUG = 000000      D.PWR = 000001          FD.DIR= 000010
CHAN09  002150R  025 CHR..U= 000165              DEFERR= 104405      D.QQAD= 000000          FD.FTN= 000001
CHAN10  002332R  025 CHR..Z= 000172              DEVIL   005114R  025 D.QUAD= 000000          FD.F11= 040000
CHAN11  002514R  025 CHR..0= 000060              DEVNTF= 104470      D.RES = 000000          FD.INS= 000010
CHAN12  002676R  025 CHR..1= 000061              DIGITS  000006R  025 D.RHO = 000000          FD.ISP= 002000
CHAN13  000022R  023 CH.AND= 000001              DIMAX = 000020      D.ROT = 000003          FD.MNT= 100000
CHAN14  000204R  023 CLRSI = ****** GX           DIMBLK  003112R  025 D.SCON= 000006          FD.OSP= 004000
CHNOPN= 104471          CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006          FD.PLC= 000004
CHRAND= 000051          CMDMAX= 000010              DIMTYP  003116R  025 D.SUB = 000000          FD.PRN= 000004
CHRBEL= 000007          CNTSIZ  005130R  025 DIRFUL= 104466          D.SUBS= 000006          FD.PSE= 010000
CHRBKR= 000133          COMBIN  005076R  025 DOMERR= 104417          D.TAKE= 000002          FD.RAH= 000001
CHRBOL= 000200          COMCNT  005126R  025 DPTERR= 104416          D.TRAN= 000004          FD.RAN= 000002
CHRBOX= 000114          COMERR= 104425              DYDCNT  003224R  025 D.UNON= 000006          FD.REC= 000001
CHRCMA= 000054          COMPTR  005122R  025 D.AND = 000000          D.$MAX= 000000          FD.RWM= 000001
CHRCOL= 000076          COMRTN  003156R  025 D.BXCT= 000007          D.$MIN= 000000          FD.SDI= 000020
CHRDEL= 000107          COMSKP  005124R  025 D.BXIO= 000007          D.$MUL= 000000          FD.SQD= 000040
CHRDIV= 000053          CORPTR  005120R  025 D.BXNL= 000007          D.$PLU= 000000          FD.TTY= 000004
CHRDLR= 000176          CS.ASS= 000200              D.BXPP= 000007          D.$RES= 000000          FD.WBH= 000002
CHRDLT= 000110          CS.BOO= 000010              D.BXPW= 000007          D.$SUB= 000000          FF.CHR= 000005
CHRDOT= 000056          CS.FLO= 000020              D.BXRL= 000007          D.1COM= 000001          FF.NV = 000003
CHREND= ****** GX       CS.IDN= 000040              D.CIRC= 000001          D.1EXP= 000002          FF.POE= 000002
CHREP = 000105          CS.ONE= 000100              D.COMB= 000001          D.1ROT= 000003          FF.RWD= 000001
CHRERR= 104431          CTEPS   000152R  025 D.COMP= 000001          EDICHR  005063R  025 FF.RWF= 000006
```

FIXLIN MACRO M1113 08-FEB-80 12:18 PAGE 48-2
SYMBOL TABLE

```
FF.SPC= 000004          F.EFN = 000050          F.VAL = 177776          L.PRNT= 000001          M.NUM = 000204
FHEADR  005012R     025 F.END = 002000          F.VBN = 000064          L.PTR = 000000          M.OUTR= 000012
FHRET   005040R     025 F.EOBB= 000032          F.VBSZ= 000060          L.PTR1= 000004          M.PITM= 000000
FILNTF= 104467          F.ERR = 000052          GNOME   003102R     025 L.SUBS= 000002          M.PRDS= 000006
FIS   = 000000          F.FACC= 000043          HASHC   005136R     025 MACHIN= 000106          M.QQAD= 000000
FISCON= 046000          F.FFBY= 000014          HASHTB  005140R     025 MAXCHN= 000014          M.QUAD= 000000
FIXEND  000236R     020 F.FLAG= 000002          HSHMSK= 177601          MNTAB = ****** GX       M.RAND= 000012
FIXERR  000442R     020 F.FLB = 000100          HSHSIZ= 000200          MNTBND= ****** GX       M.RAVL= 000000
FIXLIN  000000RG    020 F.FNAM= 000110          IASH  = ****** GX       MNTRTB= ****** GX       M.RCIP= 000000
FIXPTR  003074R     025 F.FNB = 000102          ICHAN   005050R     025 MONBLK  003120R     025 M.RDCT= 000011
FIXSHF  000444R     020 F.FNDF= 000010          ILFNAM= 104464          MONCVT  003132R     025 M.REVS= 000003
FIX1    000020R     020 F.FTYP= 000116          ILLCHN= 104540          MONLEN  003130R     025 M.ROLL= 000000
FIX2    000266R     020 F.FUN = 000010          ILLSW = 104565          MONPTR  003122R     025 M.SATA= 000001
FLAGS   004764R     025 F.FVER= 000120          INDERR= 104410          MONSFT  003126R     025 M.SCAN= 000011
FLCBUF  003250R     025 F.HEAD= 000006          INNTMP  003250R     025 MONTYP  003124R     025 M.SHAP= 000000
FLINE   005014R     025 F.HIBK= 000004          INNTM2  005016R     025 MONVAL  003134R     025 M.SIGN= 000000
FLOEXP  003270R     025 F.LBL = 000020          INPTR   000064R     025 M.ABS = 000000          M.SUBR= 000001
FLONEG  003272R     025 F.LINE= 000012          INSNAM= 104462          M.BOL = 000200          M.SYM = 000203
FLSWTB  000222R     025 F.LOC = 000004          IOERR = 104477          M.BXAV= 000007          M.SYN = 000201
FLTLEN= 000010          F.LSTC= 000004          I.CNT = 177776          M.BXCR= 000007          M.TRAN= 000012
FNINSI  005044R     025 F.LUN = 000042          I.DIM = 177772          M.BXCT= 000007          M.$ABS= 000000
FORIGN  000032R     025 F.MBCT= 000054          I.LEN = 000012          M.BXEX= 000007          M.$CEI= 000000
FO.APD= 000106          F.MBC1= 000055          I.OERR= 000062          M.BXFX= 000007          M.$FLR= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OLD = 177770          M.BXIO= 000007          M.$IDN= 000000
FO.RD = 000001          F.MON = 000002          I.PTR = 177774          M.BXLC= 000003          M.$NEG= 000000
FO.UPD= 000006          F.NIL = 000001          I.STR = 177766          M.BXNC= 000007          M.$SIG= 000000
FO.WRT= 000016          F.NRBD= 000024          LABERR= 104406          M.BXNL= 000007          M.1RDT= 000011
FPMODE  000042R     025 F.NREC= 000030          LABLPT  005042R     025 M.BXPP= 000007          M.1REV= 000003
FPP   = 000000          F.OPN = 000010          LARGBL  003162R     025 M.BXPW= 000007          M.1SCN= 000011
FPPLEN= 000004          F.OVBS= 000030          LARGCV  003176R     025 M.BXRL= 000007          NAMBUF  003240R     025
FREBLK  007400R     025 F.PREV= 000C0           LARGFL  003160R     025 M.BXTS= 000007          NB.DEV= 000200
FUZZ    000142R     025 F.PRNT= 100000          LARGLE  003174R     025 M.BXWA= 000007          NB.DIR= 000100
FXPTR1  003076R     025 F.PROT= 010000          LARGLN  003166R     025 M.CEI = 177777          NB.NAM= 000004
F$EVAL= 000100          F.RACC= 000016          LARGPT  003164R     025 M.CHR = 000202          NB.SD1= 000400
F$EVI = 000200          F.RATT= 000001          LARGSF  003172R     025 M.CMT = 000206          NB.SD2= 001000
F$EXEC= 000400          F.RCNM= 000034          LARGTY  003170R     025 M.DOUT= 000006          NB.SNM= 000040
F$FUNC= 001000          F.RCTL= 000017          LARGVA  003200R     025 M.DQAD= 000006          NB.STP= 000020
F.ACTL= 000076          F.READ= 004000          LBINC   005100R     025 M.ENCD= 000004          NB.SVR= 000010
F.ALOC= 000040          F.REDF= 000040          LBUFF   003316R     025 M.EXE = 000000          NB.TYP= 000002
F.BBFS= 000062          F.RES = 000040          LBUFLN= 000621          M.EXP = 000001          NB.VER= 000001
F.BDB = 000070          F.RET = 177772          LCOUNT  005106R     025 M.FACT= 000001          ND.DEV= 000200
F.BGBC= 000057          F.RSIZ= 000002          LENERR= 104412          M.FIL = 000177          ND.EXT= 000002
F.BITS= 177774          F.RTYP= 000000          LENTAB  000054R     025 M.FLR = 177777          ND.NAM= 000004
F.BKDN= 000026          F.R2  = 000014          LEXPTR  003100R     025 M.FMT = 000004          ND.PPN= 000100
F.BKDS= 000020          F.R3  = 000016          LFTOP   005066R     025 M.GDUP= 000003          ND.$EX= 000000
F.BKEF= 000050          F.R4  = 000020          LINC    005102R     025 M.GDWN= 000003          ND.$ES= 000020
F.BKP1= 000051          F.SEQN= 000100          LINE    005016R     025 M.GETI= 000006          ND.$NM= 000000
F.BKST= 000024          F.SPDV= 000072          LININC  005026R     025 M.IBM = 000003          ND.$NS= 000040
F.BKVB= 000064          F.SPUN= 000074          LISIZE= 000020          M.IDN = 000000          ND.$PN= 001400
F.BR  = 000020          F.STBK= 000036          LITOP   005072R     025 M.INSP= 000006          NEGSGN  005063R     025
F.CHR = 000075          F.STK = 000022          LNLNG = 104541          M.IOTA= 000000          NINPT1  003302R     025
F.CNTG= 000034          F.STOP= 000004          LOCLCT  005036R     025 M.LIT = 000205          NINPT2  003300R     025
F.CODE= 000010          F.STP = 000001          LOWCOR  000002R     025 M.LN  = 000001          NINSTK  003304R     025
F.DFNB= 000046          F.SUSP= 000002          LSTERP  004766R     025 M.MMD = 000010          NINTMP  003310R     025
F.DSPT= 000044          F.TRC = 000002          LWIDTH  000070R     025 M.NEG = 000000          NINTYP  003306R     025
F.DVNM= 000134          F.UNIT= 000136          L.BITS= 000003          M.NOT = 000000          NOTAVL= 104472
F.EFBK= 000010          F.URBD= 000020          L.CNT = 000002          M.NUB = 000006          NOTOPN= 104473
```

SYMBOL TABLE

```
NPRT1   004762R     025 OPTM14  005026R     025 RSX   = 000115          SYNERR= 104407          T.DIM = 000100
NULL    007364R     025 ORIGIN  000030R     025 RTOP    005070R   025 SYSERR= 104402          T.FUN = 010000
NUMCHN= 000020          OUTPTR  000066R     025 RT11  = 000000          SYSTAT  000000R   024 T.LIST= 000400
NUMDFL  003230R     025 OVROPR= 000000          R.BITS= 000003          SY.BAD= 000001          T.LIT = 000020
NUMEFL  003233R     025 OVRTAB= ****** GX       R.FIX = 000001          SY.BRK= 100000          T.MON = 000010
NUMESN  003236R     025 OVRTND= ****** GX       R.LCT = 000002          SY.BS = 000010          T.NIL = 000004
NUMFLG  003230R     025 OVSTRK= ****** GX       R.LN  = 000002          SY.CLR= 000040          T.NUM = 002000
NUMLEN  000062R     025 PATCH   007344R     025 R.LNUM= 000010          SY.ERP= 000020          T.OPFN= 000001
NUMNEG  003232R     025 PRECSN  000052R     025 R.NEXT= 000006          SY.NOU= 000100          T.OTHR= 040000
NUMQFL  003234R     025 RANDAC  003064R     025 R.OFF = 000000          SY.OPN= 000002          T.SOP = 000002
NUMRIG  003231R     025 RANDY   003062R     025 R.PREV= 000004          SY.PMT= 001000          T.SPOP= 000040
NUMWID  003226R     025 RANERR= 104411          R.PTR = 000000          SY.SRT= 000400          T.TMP = 000200
NUMZFL  003235R     025 RARGBL  003204R     025 R.SEQ = 000003          SY.SVC= 000004          T.VAL = 100000
NYIERR= 104432          RARGCV  003220R     025 R.VAR = 000002          SY.SVL= 000200          T.VAR = 001000
N.BLSZ= 000036          RARGFL  003202R     025 R6    =%000006          S.AB  = 177776          UPPER = 177400
N.DESC= 000020          RARGLE  003216R     025 R7    =%000007          S.ASB = 100000          UTNAME  000344R   023
N.DEV = 000032          RARGLN  003210R     025 SI    =%000005          S.BFHD= 000020          VALERR= 104413
N.DEVN= 000034          RARGPT  003206R     025 SIDAMN= 104433          S.BITS= 000002          VARBLE  007400R   025
N.DID = 000024          RARGSF  003214R     025 SIFRAM  003314R   025 S.CHR1= 000003          VARBLS  000000R   025
N.DVNM= 000032          RARGTY  003212R     025 SISTAK  006350R   025 S.FATT= 000016          VERNUM  000000R   025
N.EXT = 000014          RARGVA  003222R     025 SITOP   006346R   025 S.FDB = 000140          VRSION= 000002
N.FID = 000000          RBASE   005074R     025 SITP    007144R   025 S.FLB = 040000          WIDTH   000072R   025
N.FNAM= 000006          RCOUNR  005112R     025 SIWARN= 000200          S.FNAM= 000006          WROUND  000020R   025
N.FTYP= 000014          RCOUNT  005110R     025 SIZTAB  000044R   025 S.FNB = 000036          WSDATE  000136R   025
N.FVER= 000016          REDCNT  005070R     025 SI.BIT= 177774          S.FNBW= 000017          WSFULL= 104403
N.NAME= 000006          REDIGN  005066R     025 SI.LEN= 177772          S.FNTY= 000004          WSNAME  000074R   025
N.NEXT= 000022          REDOFF  005072R     025 SI.RET= 177772          S.FTYP= 000002          WSTIME  000132R   025
N.PPN = 000024          REENTR  003060R     025 SI.VAL= 177776          S.LEN = 000002          $FIS  = ****** G
N.STAT= 000020          REGTYP= 000000          SKPSIZ  005134R   025 S.LINK= 177774          $FPP  = ****** G
N.UNIT= 000034          RESBIT  003142R     025 SPHOLD  000004R   024 S.NFEN= 000020          $FPPLN= ****** G
OBUFLN= 001000          RESCVT  003152R     025 SPINIT  000002R   024 S.TYPE= 140000          $MACHN= ****** G
OCHAN   005052R     025 RESERR= 104440          STKINC= 000774          S.VAL = 000000          $RSTS = ****** G
OMODE   000402R     023 RESLEN  003150R     025 STOP  = 104420          TKDIMS  005072R   025 $RSX  = ****** G
OPTMP0  005066R     025 RESPTR  003140R     025 SUBCNT  003104R   025 TKFRAM  005066R   025 $RT11 = ****** G
OPTMP1  005070R     025 RESSFT  003146R     025 SUBDCT  003106R   025 TKLEFT  005070R   025 .BOOLE= 000002
OPTMP2  005072R     025 RESTYP  003144R     025 SUBPTR  003110R   025 TMODE   000403R   023 .CHARA= 000000
OPTMP3  005074R     025 RESULT  003136R     025 SYMB1   005344R   025 TMPTAG= 000166          .DEF. = 000002
OPTMP4  005076R     025 RESVAL  003154R     025 SYMF    005340R   025 TRNTAB= ****** GX       .FLOAT= 000006
OPTMP5  005100R     025 RINC    005104R     025 SYMPT1  005346R   025 TTYBUF  004140R   025 .FSRCB= ****** G
OPTMP6  005102R     025 RLTBAS  005066R     025 SYMTAB  005352R   025 TTYLEN= 000622          .INTEG= 000004
OPTMP7  005104R     025 ROTINC  005070R     025 SYMTBF= 104404          TURKEY= 104436          .MOLUN  000404R   023
OPTMP8  005106R     025 ROUND   000010R     025 SYMTOP  005350R   025 TYPE  = 000166          ...GBL= 000000
OPTMP9  005110R     025 ROVER   003312R     025 SYMTP   006272R   025 T.BOL = 020000          ...TPC= 000020
OPTM10  005112R     025 RSTS  = 000000          SYMWRN= 000054          T.CON = 004000

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
```

```
FIXLIN  MACRO M1113  08-FEB-80 12:18  PAGE 48-4
SYMBOL TABLE

OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    000466      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13044 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:17
FIXLIN,DM1:[305,301]FIXLIN/-SP=APLRX1,APLDEF,FIXLIN,END
```

```
**FILE**ID**FN
```

```
FFFFFFFFFF    NN       NN
FFFFFFFFFF    NN       NN
FF            NN       NN
FF            NN       NN
FF            NNNN     NN
FF            NNNN     NN
FFFFFFFF      NN  NN   NN
FFFFFFFF      NN  NN   NN
FF            NN    NNNN
FF            NN    NNNN
FF            NN       NN        ....
FF            NN       NN        ....
FF            NN       NN        ....
FF            NN       NN        ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:20
TABLE OF CONTENTS


```
     2-     1    APLDEF
     2-     8    A P L
     2-     9
     2-    10    A PROGRAMMING LANGUAGE
     2-    11
     2-    12    STEVE POULSEN & DAVE VANN
     2-    13    11-Jul-76
     2-    14    ROBERT GOTTLIEB & PETER HAYNES
     2-    15    2-Feb-79
     2-    16
     2-    17    V2
     2-    18
     2-    19
     2-    20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-    21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-    22
     2-    23
     2-    24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-    25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-    26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-    27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-    28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-    29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-    30    DEC.
     2-    31
     2-    32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-    33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-    34    CORPORATION.
     2-    35
     2-    36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-    37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-    38
     2-    39
     3-    46    DEFINITIONS     ASSEMBLY PARAMETERS
     4-   139    DEFINITIONS     OPERATOR OVERLAYS
     5-   268    DEFINITIONS     REGISTERS
     6-   331    DEFINITIONS     MEMORY ORGANIZATION
     7-   377    DEFINITIONS     ORDERING PSECTS
     8-   426    DEFINITIONS     SI PARSE FRAMES
     9-   474    DEFINITIONS     TOKEN DESCRIPTION BITS
    10-   515    DEFINITIONS     SI EXECUTION FRAMES
    11-   580    DEFINITIONS     SPECIAL PARSE CHARACTERS
    12-   607    DEFINITIONS     CHRTAB BITS
    12-   624    DEFINITIONS     SYSTEM STATUS BITS
    13-   640    DEFINITIONS     F-FRAMES
    14-   724    DEFINITIONS     FUNCTION LINE BLOCKS
    16-   807    DEFINITIONS     ARRAY STORAGE BLOCK
    17-   869    DEFINITIONS     LIST STORAGE BLOCK
    18-   911    DEFINITIONS     INDEXING STACK FRAME
    19-   940    DEFINITIONS     SYMBOL TABLE ENTRY
    20-   969    DEFINITIONS     CHANNEL BLOCK
    22-  1085    DEFINITIONS             RSX DATA
    23-  1110    DEFINITIONS     NAME BLOCKS
    33-  1612    DEFINITIONS     MACROS
    35-  1905    DEFINITIONS     APL CHARACTERS
    36-  1967    DEFINITIONS     ERRORS
    44-     4    FN              ENTER FUNCTION DEF. MODE [OVERLAY]
```

FN MACRO M1113 08-FEB-80 12:20
TABLE OF CONTENTS

47-	118	NEXTLN	DEFINE NEXT FUNCTION LINE
49-	225	FNEDIT	HANDLE FUNCTION EDITING
56-	659	FNOPEN	OPEN FUNCTION FOR EDITING
57-	716	EOLCHK	CHECK FOR DEL AT LINE END

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 1


     1          000000                RT11    =  0          ;NOT RT-11
     2          000000                RSTS    =  0          ;NOT RSTS/E
     3          000115                RSX     = 'M          ;RSX-11/M
     4
     5          000000                EIS     =  0          :NO EIS
     6          000000                FIS     =  0          ;NO FIS
     7          000000                FPP     =  0          ;NO FPP
     8
     9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
    10
    11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
2513                                    .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

DEFINITIONS ERRORS

```
                                .TITLE  FN
                                .IDENT  \V02.03\

                                .SBTTL  FN              ENTER FUNCTION DEF. MODE [OVERLAY]

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       This routine is branched to when 'EXEC' discovers a del
                                ;       at the beginning of a line.  'FN' must then decide if
                                ;       the user wants to open a previously defined function, or
                                ;       if he wants to define a new function.  In the latter case,
                                ;       a function header must be created, the syntax of the
                                ;       function header must be checked, and local variables must
                                ;       be identified and placed in the function header.  'FNHEAD'
                                ;       performs most of these operations.  During this process,
                                ;       a temporary val frame is placed on the SI stack and later
                                ;       removed so that an error during function definition will
                                ;       deallocate the function header block.
                                ;            Once the function header has been found or created for the
                                ;       function, a pointer to the header is placed in 'FHEADR' and
                                ;       a pointer to the current line is placed in 'FLINE'.  'LININC'
                                ;       is the current line increment (initially 1) for new function
                                ;       lines.
                                ;            When all of this is completed, control is passed to the line
                                ;       editing routines ('NEXTLN' and 'FNEDIT') which then define
                                ;       and edit function lines until a closing del is found.
                                ;
                                ; INPUTS:
                                ;       R0 -    POINTS TO FIRST CHAR PRODUCED BY 'FIXLIN'
                                ;       R1 -    COUNT OF CHARS PRODUCED BY 'FIXLIN'
                                ;--
 000000                                 .PSECT  CODE
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 45
FN              ENTER FUNCTION DEF. MODE [OVERLAY]

     34 000000  042737  010000  004764' FN::    BIC     #F.PROT,FLAGS   ;CLEAR THE 'LOCKED' INDICATOR
     35 000006                           PUSH    <R0,R1>         ;SAVE LINE POINTER AND COUNT
     36 000012  010004                   MOV     R0,R4           ;COPY POINTER
     37 000014  010103                   MOV     R1,R3           ;COPY COUNT
     38 000016                   1$:     CALL    GETNB           ;GET NON-BLANK CHAR
     39 000022  002421                   BLT     4$              ;NONE LEFT
     40 000024  122702  000107           CMPB    #CHRDEL,R2      ;NO, IS IT A DEL?
     41 000030  001407                   BEQ     3$              ;IT SHOULD BE
     42 000032  122702  000271           CMPB    #CHRPDL,R2      ;HOW ABOUT PROTECT DELS?
     43 000036  001401                   BEQ     2$              ;THEY'RE ALSO OK
     44 000040  104402                   SYSERR                  ;NO OPENING DEL? HOW DID WE GET HERE?
     45 000042  052737  010000  004764' 2$:     BIS     #F.PROT,FLAGS   ;DON'T FORGET WHEN YOU CLOSE DEF
     46 000050                   3$:     CALL    GETNB           ;GET NON-BLANK CHAR
     47 000054  002404                   BLT     4$              ;NONE LEFT
     48 000056  132762  000010  000000G        BITB    #C.IDEN,CHRTAB(R2);IS IT AN IDENTIFIER?
     49 000064  001001                   BNE     5$              ;YES, PERHAPS FUNCTION'S NAME??
     50 000066  104405                   4$:     DEFERR          ;ELSE DEFN ERROR
     51 000070                   5$:     CALL    SYMBOL          ;GET FUNCTION'S NAME
     52 000074                   6$:     CALL    GETNB2          ;GET NON-BLANK CHAR
     53 000100  002772                   BLT     4$              ;NONE LEFT, ERROR
     54                                  ;+
     55                                  ; NOW WE HAVE (POSSIBLY) THE NAME OF THE FUNCTION TO EDIT.
     56                                  ; IT IS STILL NECESSARY TO CHECK IF THIS IS A FUNCTION EDIT
     57                                  ; OR A FUNCTION DEFINITION.
     58                                  ;-
     59 000102  005037  005044' 7$:      CLR     FNINSI          ;ASSUME WE CAN EDIT FUNCTION
     60 000106  011000                   MOV     @R0,R0          ;THIS IS FUNCTION EDIT, GET POINTER
     61 000110  001404                   BEQ     8$              ;NOTHING, IS NEW
     62 000112  032760  040000  000002       BIT     #S.FLB,R.LN(R0) ;DOES NAME POINT TO FUNCTION HEADER??
     63 000120  001460                   BEQ     FND             ;NO, CAN'T EDIT A VARIABLE!!!!
     64
     65 000122  013701  003314' 8$:      MOV     SIFRAM,R1       ;POINT TO TOP FRAME
     66 000126  000411                   BR      10$             ;AND START SEARCH
     67
     68 000130  032761  001000  000002 9$:     BIT     #F$FUNC,F.FLAG(R1)      ;F-FRAME?
     69 000136  001405                   BEQ     10$             ;NOPE
     70 000140  020061  000010           CMP     R0,F.FUN(R1)    ;IS IT THE FUNCTION WE WANT TO EDIT?
     71 000144  001002                   BNE     10$             ;NO
     72 000146  005237  005044'          INC     FNINSI          ;INCREMENT COUNT OF OCCURANCES
     73 000152  011101           10$:    MOV     @R1,R1          ;POINT TO NEXT FRAME
     74 000154  001365                   BNE     9$              ;AND KEEP LOOKING IF MORE FRAMES
     75
     76 000156  122702  000073           CMPB    #CHRLBR,R2      ;'['?
     77 000162  001010                   BNE     11$             ;NO, THEN THIS IS A FUNCTION DEF!
     78 000164  005700                   TST     R0              ;ANYTHING THERE?
     79 000166  001737                   BEQ     4$              ;NO, YOU CAN'T DO THIS
     80 000170                           CALL    FNOPEN          ;OPEN THE FUNCTION
     81 000174  062706  000004           ADD     #4,SP           ;CLEAN UP POINTERS ON STACK
     82 000200  000137  000642'          JMP     FNEDIT          ;AND CALL FUNCTION EDITOR
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 46
FN              ENTER FUNCTION DEF. MODE [OVERLAY]


                                        ;+
                                        ; THE USER HAS NOT SPECIFIED A LINE TO EDIT, SO WE HAVE
                                        ; TO DETERMINE IF HE IS DEFINING A FUNCTION OR JUST WANTS
                                        ; TO OPEN A FUNCTION AT ITS LAST LINE.
                                        ;-
000204  042737  000004  004764' 11$:     BIC     #F.STOP,FLAGS   ;CLEAR STOP FLAG (DEL NOT FOUND YET)
000212  122702  000107                  CMPB    #CHRDEL,R2      ;DEL?
000216  001406                          BEQ     12$             ;YES, OK
000220  122702  000271                  CMPB    #CHRPDL,R2      ;PERHAPS A PRO. DEL?
000224  001011                          BNE     14$             ;NOPE
000226  052737  010000  004764'        BIS     #F.PROT,FLAGS   ;PROTECT ME, PLEASE!
000234  052737  000004  004764' 12$:    BIS     #F.STOP,FLAGS   ;YES, SET DEFINITION MODE EXIT FLAG
000242                          13$:    CALL    GETNB           ;GET NON-BLANK CHAR
000246  002403                          BLT     15$             ;NONE LEFT
000250  122702  000015          14$:    CMPB    #CHR.CR,R2      ;END OF LINE?
000254  001002                          BNE     FND             ;NO, SO HE IS DEFINING NEW FUNCTION
000256  005700                  15$:    TST     R0              ;Defining new niladic function?
000260  001007                          BNE     FNG             ;He really is opening old function
000262                          FND::   POP     <R1,R0>         ;GET BACK INPUT POINTERS
                                        ;+
                                        ; USER IS DEFINING A NEW FUNCTION.  ANALYZE HEADER, FILL
                                        ; IN THE NEW HEADER BLOCK, AND START TO DEFINE THE FUNCTION.
                                        ;-
000266                          FNRDF:  CALL    CKFNSI          ;Is it on the SI?
000272                                  CALL    FNHEAD          ;ANALYZE FUNCTION HEADER
000276  104405                          DEFERR                  ;Definition error
                                        ;+
                                        ; USER IS OPENING OLD FUNCTION
                                        ;-
000300                          FNG:    CALL    FNOPEN          ;Open the Function for Editing
000304  013705  003314'                 MOV     SIFRAM,SI       ;POINT TO E-FRAME
000310  005065  000004                  CLR     F.LSTC(SI)      ;THERE IS NO  LONGER A CODESTRING
000314  016505  000006                  MOV     F.HEAD(SI),SI   ;POINT TO WHERE CODESTRING GOES
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 47
NEXTLN          DEFINE NEXT FUNCTION LINE


                                        .SBTTL  NEXTLN          DEFINE NEXT FUNCTION LINE

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       'NEXTLN' prompts the user, then accepts a line of input.
                                        ;       The line is checked for a label and then LEX is called
                                        ;       to convert the line into a codestring.  The codestring,
                                        ;       line number, and other junk is then copied into a
                                        ;       function line block which is then linked into the circular
                                        ;       list of finction lines.  A duplicate line is deleted if
                                        ;       necessary.
                                        ;           'LINE' should contain the last line # defined, and
                                        ;       'LININC' is the increment to be added to 'LINE' to get the
                                        ;       number of the new line to be defined.
                                        ;           If a closing del is found, 'FNEND' is called so that
                                        ;       the function can be cleaned up and closed.
                                        ;--
000320                                  .PSECT  CODE
                                        .ENABL LSB

000320                                  NEXTLN::
000320                                          LDF     LINE,AC0        ;GET CURRENT LINE #
000330                                          ADDF    LININC,AC0      ;ADD LINE INCREMENT
000340                                          STF     AC0,LINE        ;STORE NEW LINE NUMBER
000350                                  SAMELN::
000350  032737  000400  004764'                 BIT     #F$EXEC,FLAGS   ;DEFINING FUNCTION BY EXECUTE OP?
000356  001024                                  BNE     SAMLN1          ;YES, SO NO PROMPT!
000360  022737  000256' 005050'                 CMP     #CHAN00,ICHAN   ;INPUT FROM TTY?
000366  001020                                  BNE     SAMLN1          ;NO, SO NO PROMPT
000370  012700  000073                         MOV     #CHRLBR,R0      ;GET A LEFT BRACKET
000374                                          CALL    OUTCHR          ;PRINT IT
000400  010537  003274'                         MOV     SI,ALLINT       ;PRINT NUMBER WITH CLOSE PACKING
000404                                          LDF     LINE,AC1        ;PREVENT 'RANDOM'' LINE NOS FOR BLANK LINES
000414                                          CALL    FLOUT           ;PRINT CURRENT LINE NUMBER
000420                                          QPRINT  RBRTXT          ;PRINT ']   '
000420  012701  000000G                         MOV #RBRTXT,R1          ;POINT TO RBRTXT
000430                                  SAMLN1::
000430                                          CALL    GETLIN          ;GET A LINE OF INPUT
000434  032737  000400  004764'                 BIT     #F$EXEC,FLAGS   ;EXECUTE OP CALL US?
000442  001002                                  BNE     1$              ;YES, SO LINE IS ALREADY FIXED
000444                                          CALL    FIXLIN          ;FIX IT UP
000450  010004                          1$:     MOV     R0,R4           ;COPY CHARACTER POINTER
000452  010103                                  MOV     R1,R3           ;COPY CHARACTER COUNT
000454                                  2$:     CALL    GETNB           ;GET NON-BLANK CHAR
000460  002467                                  BLT     NEXTER          ;NONE LEFT, ERROR
000462                                  SAMLN2::
000462  022702  000073                         CMP     #CHRLBR,R2      ;LEFT BRACKET?
000466  001465                                  BEQ     FNEDIT          ;YES, CALL THE LINE EDITOR
```

```
                                        ;+
                                        ; WE NOW KNOW THAT THE USER DOESN'T WANT TO EDIT THE LINE.
                                        ; (CHANGE THE LINE NUMBER, LIST A LINE, ETC.)  NOW CHECK IF
                                        ; HE WANTS TO DELETE THE LINE OR IF HE JUST TYPED A STANDARD
                                        ; FUNCTION LINE.
                                        ;-
000470  022702  000107          3$:     CMP     #CHRDEL,R2      ;IS CHAR A DEL?
000474  001406                          BEQ     4$              ;YES, IS GOOD
000476  122702  000271                  CMPB    #CHRPDL,R2      ;A PROTECT DEL?
000502  001015                          BNE     6$              ;NOPE
000504  052737  010000  004764'        BIS     #F.PROT,FLAGS   ;SET FOR LOCK
000512                          4$:     CALL    GETNB           ;GET NON-BLANK CHAR
000516  002403                          BLT     5$              ;NONE LEFT
000520  122702  000015                  CMPB    #CHR.CR,R2      ;CR AFTER DEL?
000524  001045                          BNE     NEXTER          ;NO, ERROR
000526                          5$:     CALL    FNEND           ;Finish up the the function
000532  000137  000000G                 JMP     EXEC            ;HE WANTS TO EXIT FROM FN DEFINITION

000536  122702  000015          6$:     CMPB    #CHR.CR,R2      ;NULL LINE?
000542  001702                          BEQ     SAMELN          ; YES, TRY AGAIN
000544  005037  005042'                 CLR     LABLPT          ;WE HAVE NO LABEL YET
000550                                  TSTF    LINE            ;IS THIS THE FN HEADER?
000560                                  CFCC                    ;GET CONDITIONS
000564  001012                          BNE     7$              ;GOOD! NOT THE HEADER
000566  052737  000040  004764'        BIS     #F.REDF,FLAGS   ;BLECH...TRYING TO REDEFINE THE FN HEADER
000574                                  PUSH    FHEADR          ;PUSH PTR TO FN HEADER
000600  010400                          MOV     R4,R0           ;COPY POINTERS
000602  010301                          MOV     R3,R1           ;FOR LEX
000604  005300                          DEC     R0              ;BUT BACK UP
000606  005201                          INC     R1              ;ONE CHAR ('CAUSE WE ALREAD READ IT!)
000610  000626                          BR      FNRDF           ;Redefining the Header!

000612                          7$:     CALL    FNLINE          ;Get a function line
000616  104405                          DEFERR                  ;Definition Error
000620  032737  002004  004764'        BIT     #F.STOP+F.END,FLAGS     ;HIT A DEL OR END OF EXECUTION STRING?
000626  001634                          BEQ     NEXTLN          ;No, go get another line
000630                          8$:     CALL    FNEND           ;Finish up the function
000634  000137  000000G                 JMP     EXEC            ;ELSE EXIT FROM FN DEF MODE

                                        .DSABL LSB

000640  104407                  NEXTER: SYNERR                  ;ERROR IN FUNCTION LINE DEFINITION
```

```
                                .SBTTL  FNEDIT          HANDLE FUNCTION EDITING

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'FNEDIT' IS CALLED WHEN A LEFT BRACKET IS FOUND AT THE
                                ;       FRONT OF A LINE OF A FUNCTION DEFINITION.  THIS COULD
                                ;       MEAN SEVERAL THINGS:  LIST THE FUNCTION, EDIT A LINE,
                                ;       OR JUST CHANGE THE LINE NUMBER SEQUENCE AND INCREMENT.
                                ;       THIS ROUTINE ALSO DETECTS A DEL AT THE END OF A LINE
                                ;       AND CALLS 'FNEND' IF NECESSARY.
                                ;
                                ; INPUTS:
                                ;       R2 -    CURRENT CHAR (LEFT BRACKET)
                                ;       R3 -    COUNT OF CHARS LEFT IN LINE
                                ;       R4 -    POINTER TO NEXT INPUT CHAR
                                ;--
000642                          .PSECT  CODE

                                .ENABL  LSB             ;3 CHEERS FOR LOCAL SYMBOLS!?!

000642                  FNEDIT: CALL    GETNB           ;GET NON-BLANK CHAR
000646  002415                  BLT     1$              ;NONE LEFT
000650  132762  000020  000000G       BITB    #C.NUM,CHRTAB(R2);NUMERIC?
000656  001012                  BNE     12$             ;YES, GO GET LINE NUMBER
000660  122702  000056          CMPB    #CHRDOT,R2      ;DOT
000664  001407                  BEQ     12$             ;YES, IT IS NUMERIC!
000666  122702  000114          CMPB    #CHRBOX,R2      ;QUAD?
000672  001406                  BEQ     2$              ;YES, HE WANTS TO LIST SOMETHING
000674  122702  000110          CMPB    #CHRDLT,R2      ;DELTA?
000700  001517                  BEQ     FNEDDL          ;Yes, line delete

000702  104407          1$:     SYNERR                  ;ELSE BAD EDIT SYNTAX
000704  000137  001306'  12$:    JMP     FNEDT1          ;So go get the line number

                                ;+
                                ; USER WANTS TO LIST ALL OR PART OF THE FUNCTION.
                                ;-

000710                  2$:     CALL    GETNB           ;GET NON-BLANK CHAR
000714  002772                  BLT     1$              ;NONE LEFT
000716  132762  000020  000000G       BITB    #C.NUM,CHRTAB(R2);IS CHAR NUMERIC?
000724  001003                  BNE     8$              ;YES, GET NUMBER
000726  122702  000056          CMPB    #CHRDOT,R2      ;IS IT A DOT?
000732  001072                  BNE     10$             ;NO, SO CAN'T BE NUMERIC
000734                  8$:     CALL    NUMBER          ;GET LINE AT WHICH TO START
000740                  3$:     CALL    GETNB2          ;GET NON-BLANK CHAR
000744  002756                  BLT     1$              ;NONE LEFT, ERROR
000746  122702  000047  4$:     CMPB    #CHRRBR,R2      ;NUMBER END WITH ']'?
000752  001353                  BNE     1$              ;NO, BAD EDIT SYNTAX
000754                          CALL    EOLCHK          ;YES, CLEAN UP END OF LINE
000760  013700  005012'          MOV     FHEADR,R0       ;POINT TO FN HEADER
000764  010001                  MOV     R0,R1           ;COPY POINTER
000766  016101  000006  5$:     MOV     R.NEXT(R1),R1   ;POINT TO NEXT LINE BLOCK
000772  020001                  CMP     R0,R1           ;DID WE GO ALL THE WAY AROUND?
000774  001411                  BEQ     7$              ;YES, LINE NOT FOUND (TOO BIG!)
000776                          CMPF    R.LNUM(R1),AC0  ;CHECK LINE NUMBERS
001006                          CFCC                    ;CHECK FEDERAL COMMUNICATIONS COMMISSION
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 49-1
FNEDIT          HANDLE FUNCTION EDITING

    282 001012  002765                          BLT     5$              ;DON'T START LISTING YET
    283
    284 001014                          6$:     CALL    FNLIST          ;LIST THE REST OF THE FUNCTION
    285 001020  013700  005012'         7$:     MOV     FHEADR,R0       ;POINT TO FN HEADER
    286 001024  016001  000004                  MOV     R.PREV(R0),R1   ;POINT TO LAST LINE
    287 001030  010137  005014'                 MOV     R1,FLINE        ;SAVE POINTER TO LINE
    289 001034                                  LDF     R.LNUM(R1),AC0  ;GET LAST LINE #
    290 001044                                  STF     AC0,LINE        ;MAKE IT CURENT LINE #
    291 001054                                  LDF     #^F1.0,AC0      ;GET A 1.0
    292 001064                                  STF     AC0,LININC      ;MAKE IT LINE INCREMENT
    299 001074  032737  000004  004764' FNEDTX: BIT     #F.STOP,FLAGS   ;FIND CLOSING DEL ANYWHERE?
    300 001102  001002                          BNE     9$              ;YES, STOP
    301 001104  000137  000320'                 JMP     NEXTLN          ;ELSE GO GET NEXT LINE NUMBER
    302 001110                          9$:     CALL    FNEND           ;Finish up the function
    303 001114  000137  000000G                 JMP     EXEC            ;EXIT FN DEF MODE.....
    304
    305                                 ;
    306                                 ;LIST ALL OF THE FUNCTION.
    307                                 ;
    308
    309 001120  122702  000047          10$:    CMPB    #CHRRBR,R2      ;MAKE SURE THERE IS A CLOSING RIGHT BRACKET
    310 001124  001266                          BNE     1$              ;THERE ISN'T, SYNTAX ERROR
    311 001126                                  CALL    EOLCHK          ;CLEAN UP END OF LINE
    312 001132  013701  005012'                 MOV     FHEADR,R1       ;POINT TO FN HEADER
    313 001136  000726                          BR      6$              ;LIST THE FUNCTION AND EXIT
    314                                         .DSABL  LSB             ;DISABLE THE VETERANS
    315
```

```
                                        ;
                                        ;       A DELTA HAS BEEN FOUND ... (DELETE THIS LINE)
                                        ;
001140                          FNEDDL: CALL    GETNB           ;GET NON-BLANK CHAR
001144  002457                         BLT     4$              ;NONE LEFT
001146  132762  000020  000000G        BITB    #C.NUM,CHRTAB(R2);IS CHAR NUMERIC?
001154  001003                         BNE     1$              ;YES, GET NUMBER
001156  122702  000056                 CMPB    #CHRDOT,R2      ;PERIOD?
001162  001344                         BNE     FNEDTX          ;NO, SO EOL
001164                          1$:     CALL    NUMBER          ;DIGITS, ANYONE?
001170                          2$:     CALL    GETNB2          ;GET NON-BLANK CHAR
001174  002773                         BLT     1$              ;NONE LEFT, ERROR
001176  122702  000047          3$:     CMPB    #CHRRBR,R2      ;ENDING ']' ?
001202  001040                         BNE     4$              ;WRONGO !
001204                                  CALL    EOLCHK          ;CLEAR UP EOL
001210                                  STF     AC0,LINE        ;IS IT ZERO?
001220                                  CFCC                    ;LOAD FPP COND CODE
001224  001723                         BEQ     FNEDTX          ;YOU CANNA DELETE LINE 0
001226                                  CALL    LINKIL          ;DIE, LINE !
001232                                  LDF     LINE,AC0        ;Get the current line #
001242                                  SUBF    #^F1.0,AC0      ;POINT TO THE DELETED LINE - 1
001252                                  STF     AC0,LINE        ;AND THE NEXT LINE IS ...
001262                                  LDF     #^F1.0,AC1      ;GET A ONE
001272                                  STF     AC1,LININC      ;   FOR THE INCREMENT
001302  000674                         BR      FNEDTX          ;AND TRY AGAIN

001304  104407                  4$:    SYNERR                  ;SYNTAX ERROR
```

```
                                        ;+
                                        ; A NUMBER HAS BEEN FOUND...(IT'S A LINE NUMBER).. DOES USER
                                        ; WANT TO DEFINE A DIFFERENT LINE OR DOES HE WANT SOMETHING
                                        ; SPECIAL??
                                        ;-
001306                          FNEDT1: CALL    NUMBER          ;GET THE LINE NUMBER
001312                                  STF     AC0,LINE        ;CHECK LINE NUMBER
001322                                  CFCC                    ;CHECK NUMBER
001326  100447                          BMI     6$              ;NEGATIVE LINE NUMBERS NOT ALLOWED
001330                                  LDF     #^F1.0,AC1      ;GET NEW LINE INCREMENT
001340  105737  003230'                  TSTB    NUMDFL          ;DID HE USE A DECIMAL POINT?
001344  001411                          BEQ     2$              ;NOPE
001346  113700  003231'                  MOVB    NUMRIG,R0       ;GET COUNT OF DECIMAL DIGITS
001352  001406                          BEQ     2$              ;NONE, ALL DONE
001354                          1$:     DIVF    #^F10.0,AC1     ;YES, FIND NEW LINE INCREMENT
001364                                  SOB     R0,1$           ;FIND POWER OF 10 HE SPECIFIED
001370                          2$:     STF     AC1,LININC      ;AND SAVE NEW LINE INCREMENT
001400                                  CALL    GETNB2          ;GET NON-BLANK CHAR
001404  002420                          BLT     6$              ;NONE LEFT, ERROR
001406  122702  000047                  CMPB    #CHRRBR,R2      ;RIGHT BRACKET?
001412  001012                          BNE     4$              ;NO(?)
001414                                  CALL    GETNB           ;GET NON-BLANK CHAR
001420  002412                          BLT     6$              ;NONE LEFT, ERROR
001422  122702  000015                  CMPB    #CHR.CR,R2      ;JUST A NEW LINE NUMBER?
001426  001002                          BNE     3$              ;NO
001430  000137  000350'                  JMP     SAMELN          ;YES, LET USER CHANGE THAT LINE

001434  000137  000462'          3$:     JMP     SAMLN2          ;AND DEFINE THE NEW LINE
                                        ;+
                                        ; USER WANTS TO SOME SPECIAL EDITING...(LINE EDITING OR
                                        ; LIST AND CHANGE LINE)
                                        ;-
001440  122702  000114          4$:     CMPB    #CHRBOX,R2      ;QUAD?
001444  001401                          BEQ     7$              ;YES, GOOD
001446  104407                  6$:     SYNERR                  ;NO, THEN BAD LINE SYNTAX
001450                          7$:     CALL    GETNB           ;GET NON-BLANK CHAR
001454  002774                          BLT     6$              ;NONE LEFT, ERROR
001456  013700  005012'                  MOV     FHEADR,R0       ;POINT TO FN HEADER
001462                                  LDF     AC0,AC2         ;Copy the line value
001470  010001                          MOV     R0,R1           ;COPY POINTER
001472                          8$:     LDF     R.LNUM(R1),AC1  ;Get the line #
001502                                  CALL    CMP.F           ;COMPARE LINE # W/ AC0
001506  001406                          BEQ     9$              ;THIS IS THE LINE TO LIST!
001510  016101  000006                  MOV     R.NEXT(R1),R1   ;ELSE GET NEXT LINE
001514  020100                          CMP     R1,R0           ;ALL THE WAY AROUND?
001516  001365                          BNE     8$              ;NO, KEEP LOOKING FOR LINE
001520  000137  001074'                  JMP     FNEDTX          ;ELSE SKIP THE WHOLE THING
```

```
                                        ;+
                                        ; LIST THE LINE THEN ALLOW USER TO CHANGE IT
                                        ;-
001524  122702  000047          9$:     CMPB    #CHRRBR,R2      ;END WITH RBR?
001530  001024                          BNE     10$             ;NO?  MUST BE LINE EDIT
001532                                  CALL    FLNLST          ;LIST THE LINE HE WANTS TO SEE
001536                                  CALL    NEWLIN          ;Print <CR><LF>, too...
001542                                  LDF     LINE,AC0        ;GET THE LINE NUMBER
001552                                  SUBF    LININC,AC0      ;FUDGE IT SO HE CAN CHANGE SAME LINE
001562                                  STF     AC0,LINE        ;BACK UP THE LINE NUMBER
001572                                  CALL    EOLCHK          ;END OF THE LINE?
001576  000137  001074'                 JMP     FNEDTX          ;AND GO LET HIM REDEFINE LINE
                                        ;+
                                        ; SPECIAL LINE EDITING.  GET NUMBER OF INITIAL SPACES THEN
                                        ; ALLOW USER TO CHANGE LINE
                                        ;-
001602                          10$:    PUSH    R1              ;SAVE POINTER TO LINE
001604  132762  000020  000000G         BITB    #C.NUM,CHRTAB(R2);IS CHARACTER NUMERIC?
001612  001715                          BEQ     6$              ;NO, THEN WHAT DOES HE WANT???
001614                                  CALL    NUMBER          ;GET NUMBER OF SPACES TO PRINT
001620                          11$:    CALL    GETNB2          ;GET NON-BLANK CHAR
001624  002710                          BLT     6$              ;NONE LEFT, ERROR
001626  122702  000047          12$:    CMPB    #CHRRBR,R2      ;RIGHT BRACKET?
001632  001305                          BNE     6$              ;NO, SCREW HIM?!?
001634                                  CALL    EOLCHK          ;CHECK FOR GARBAGE AT LINE END
001640  022737  000256' 005050'         CMP     #CHAN00,ICHAN   ;INPUT FROM TTY?
001646  001401                          BEQ     15$             ;YES
001650  104407                          SYNERR                  ;IF NOT, EDIT IS NOT LEGAL
001652                          15$:    STCFI   AC0,EDICNT      ;GET INITIAL SPACES
001662  003003                          BGT     16$             ;Value is acceptable
001664  013737  000250' 005064'         MOV     C.WIDR+CHAN00,EDICNT    ;Maximum value acceptable
001672  011601                  16$:    MOV     (SP),R1                 ;GET LINE POINTER AGAIN
001674  012702  000256'                 MOV     #CHAN00,R2              ;Point to channel
001700  012762  010401  177756         MOV     #<C.OPN+C.NOBF+C.FRC>,C.SBIT(R2)  ;Set the bits
001706  016262  177764  177760         MOV     C.CPTR(R2),C.BUFF(R2)   ;Get pointer to the buffer
001714  016262  177766  177762         MOV     C.CCNT(R2),C.BUFL(R2)   ;Get the buffer size
001722  005062  177770                 CLR     C.WIDC(R2)                      ;No width checking
001726  005062  177772                 CLR     C.WIDR(R2)                      ;  done this time
001732  013703  005052'                MOV     OCHAN,R3                ;Save the current channel
001736  010237  005052'                MOV     R2,OCHAN                ;Make the channel the execute one
001742                                  CALL    FLNLST                  ;Fake printing the line
001746                                  PUSH    C.CCNT+CHAN00           ;Get the space remaining
001752  005004                          CLR     R4                      ;Clear the multi-line literal flag
001754  013702  000236'                 MOV     C.BUFF+CHAN00,R2        ;Point to beginning of the line
                                                                ;Check for multi-line literal
001760  020237  000242'          41$:    CMP     R2,C.CPTR+CHAN00        ;End of the line?
001764  103004                          BHIS    42$                     ;Yes
001766  122227  000015                  CMPB    (R2)+,#CHR.CR           ;Is the character a carriage return?
001772  001372                          BNE     41$                     ;No, Just keep looping
001774  005204                          INC     R4                      ;Yes, Set the multi-line literal flag

001776  005037  000234'          42$:    CLR     C.SBIT+CHAN00           ;Close the channel
002002  010337  005052'                 MOV     R3,OCHAN                ;Restore the current channel
002006                                  CALL    TTINIT                  ;Set terminal all pretty
002012                                  CALL    FLNLST                  ;CREATE INPUT LINE
002016  013703  000250'                 MOV     C.WIDR+CHAN00,R3        ;Current line size
002022  163703  000246'                 SUB     C.WIDC+CHAN00,R3        ;   minus space remaining
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 52-1
FNEDIT          HANDLE FUNCTION EDITING

    492 002026  163716  000244'                 SUB     C.CCNT+CHANOO,(SP)      ;Compute difference in line sizes
    493 002032                                  CALL    NEWLIN                  ;Output <CR><LF>
    494 002036  005704                          TST     R4                      ;Was there a multi-line literal?
    495 002040  001402                          BEQ     43$                     ;No, Great!
    496 002042  000137  000350'                 JMP     SAMELN                  ;Yes, Can't edit it, Re-prompt
    497 002046  005726                  43$:    TST     (SP)+                   ;Same size as original?
    498 002050  001401                          BEQ     17$                     ;No, line is too long
    499 002052  104541                          LNLNG                           ;Line Tooooooo Looooonnnnnggggg
    500 002054  023703  005064'         17$:    CMP     EDICNT,R3               ;IS DESIRED COUNT BIGGER ?
    501 002060  003002                          BGT     18$                     ;NO, LEAVE AS IS
    502 002062  013703  005064'                 MOV     EDICNT,R3               ;YES, SUBSTITUTE SIZE OF LINE
    503                                         ;
    504                                 ;HERE, WE LOAD THE SPACES AND CALL GETLIN
    505                                         ;
    506 002066  012700  000040          18$:    MOV     #CHRSPA,R0      ;The character is a Space
    507 002072  005303                          DEC     R3              ;ADJUST FOR LINE
    508 002074  003404                          BLE     20$             ;DON'T NEED ANY SPACES
    509 002076                          19$:    CALL    OUTCHR          ;Output a space
    510 002102                                  SOB     R3,19$          ;LOOP FOR LEADING SPACES
    511 002106                          20$:    CALL    GETLIN          ;GET THE LINE FROM THE TERMINAL
    512 002112                                  CALL    FIXLIN          ;REDUCE BACKSPACES, ETC., ETC.
    513 002116  010004                          MOV     R0,R4           ;R4 -> LINE
    514 002120  010103                          MOV     R1,R3           ;R3 = # OF CHARS IN THE LINE
    515 002122                                  POP     R1              ;Pop the line pointer
    516 002124                                  CALL    FLNLST          ;List the function line again
```

```
                                        ;++
                                        ;       NOW, WE FIX UP FOR INSERTION/DELETION
                                        ;
                                        ;        (R4,R3) - REFERS TO THE EDIT BUFFER
                                        ;        (R0,R1) - REFERS TO THE FUNCTION BUFFER
                                        ;
                                        ;        STACK CONFIGURATION:
                                        ;       +0: REMAINING SPACE IN THE BUFFER
                                        ;       +2: CURRENT CHARACTER POSITIONS IN LINE
                                        ;       +4: POINTER TO THE START OF THE BUFFER
                                        ;       +6: LENGTH OF THE BUFFER
                                        ;--

002130  013700  000236'                         MOV     C.BUFF+CHAN00,R0;GET BUFFER POINTER
002134  013701  000240'                         MOV     C.BUFL+CHAN00,R1;TOTAL BUFFER LENGTH
002140                                          PUSH    <R1,R0>         ;SAVE BUFFER LENGTH, POINTER
002144  005046                                  CLR     -(SP)           ;CURRENT LINE POSITION
002146                                          PUSH    C.CCNT+CHAN00   ;SPACE REMAINING IN THE BUFFER
002152  161601                                  SUB     (SP),R1         ;THIS IS THE LENGTH OF THE LINE
002154  012737  177777  005064'                 MOV     #-1,EDICNT      ;NO COLUMN AS YET
002162  020103                                  CMP     R1,R3           ;WHICH IS LONGER ?
002164  100001                                  BPL     21$             ;OK
002166  010103                                  MOV     R1,R3           ;IGNORE EXTRA GARBAGE

002170                                  21$:    GETC    33$             ;GET A CHAR FROM EDIT LINE
002170  005303                                  DEC R3                          ;ANY CHARACTERS LEFT?
002172  002527                                  BLT 33$                 ;NO, BRANCH TO 33$
002174  112402                                  MOVB (R4)+,R2                   ;YES, GET NEXT CHAR
002176  100510                                  BMI     30$             ;IGNORE OVERSTRIKES
002200  120227  000015                          CMPB    R2,#CHR.CR      ;A <CR>?
002204  001522                                  BEQ     33$             ;YES, NO MORE
002206  120227  000057                          CMPB    R2,#CHR.SL      ;A "/" (SLASH) ?
002212  001026                                  BNE     24$             ;NO, TRY AGAIN
                                        ;+
                                        ;  Process slashes
                                        ;-
                                                INC     (SP)            ;YES, SO ADD A SPACE TO THE FREE AREA
                                                DEC     R1              ;LINE LENGTH LESS ONE
                                                BLE     38$             ;TOO LITTLE LEFT, WE'RE FINISHED
                                                MOV     R0,R2           ;SAVE POINTER
                                                INC     R2              ;   FOR USE IN THE LOOP
                                        22$:    PUSH    <R1,R0>         ;SAVE REGS
                                        23$:    MOVB    (R2)+,(R0)+     ;***BLOCK MOVE***
                                                SOB     R1,23$          ;(EXPLITIVE DELETED, DAMN IT!)
                                                POP     <R0,R1>         ;RESTORE THE REGISTERS
002244  121027  000010                          CMPB    (R0),#CHR.BS    ;IS THE NEXT CHAR A BACKSPACE?
002250  001347                                  BNE     21$             ;NO, NO, OF COURSE NOT
002252  012702  000002                          MOV     #2,R2           ;YES
002256  060216                                  ADD     R2,(SP)         ;2 MORE SPACES IN THE BUFFER
002260  160201                                  SUB     R2,R1           ;2 LESS IN THE LINE
002262  003517                                  BLE     38$             ;EMPTY, SO END
002264  060002                                  ADD     R0,R2           ;POINT TO OVER FOR SHIFT
002266  000757                                  BR      22$             ;AND TRY AGAIN
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 54
FNEDIT          HANDLE FUNCTION EDITING

    569                                 ;+
    570                                 ;  Process Letters
    571                                 ;-
    572 002270  132762  000010  000000G 24$:    BITB    #C.IDEN,CHRTAB(R2);A LETTER ?
    573 002276  001410                         BEQ     25$             ;NO
    574 002300  162702  000140                 SUB     #CHR..A-1,R2    ;YES, CONVERT TO NUMBER
    575 002304  100445                         BMI     30$             ;DELTAS ARE IGNORED
    576 002306                                 PUSH    R2              ;MULTIPLY TIME, KIDDIES
    577 002310  006302                         ASL     R2              ;R2 * 2.
    578 002312  006302                         ASL     R2              ;R2 * 4.
    579 002314  062602                         ADD     (SP)+,R2        ;R2 * 5.
    580 002316  000407                         BR      26$             ;AND ADD SPACES
    581                                 ;+
    582                                 ;  Process Digits
    583                                 ;-
    584 002320  132762  000040  000000G 25$:    BITB    #C.DIG,CHRTAB(R2);A DIGIT ?
    585 002326  001434                         BEQ     30$             ;NO
    586 002330  162702  000060                 SUB     #CHR..0,R2      ;YES, ASCII -> NUMBER
    587 002334  003431                         BLE     30$             ;IGNORE 0
    588
    589 002336  160216                  26$:   SUB     R2,(SP)         ;LESS SPACE REMAINING
    590 002340  003443                         BLE     32$             ;TOO MUCH, DIE
    591 002342  005737  005064'                TST     EDICNT          ;ALREADY SPACIFIED ?
    592 002346  100003                         BPL     27$             ;YES
    593 002350  016637  000002  005064'        MOV     2(SP),EDICNT    ;NO, GIVE THE START COLUMN
    594 002356                          27$:   PUSH    <R1,R2>         ;SAVE THE REGISTERS
    595 002362  060100                         ADD     R1,R0           ;POINT PAST THE ENDF
    596 002364  060002                         ADD     R0,R2           ;(A LITTLE OLDE ENGLISH, THERE)
    597 002366  114042                  28$:   MOVB    -(R0),-(R2)     ;***BLOCK TRANSFER***
    598 002370                                 SOB     R1,28$
    599 002374  011601                         MOV     (SP),R1         ;GET # OF SPACES
    600 002376  010216                         MOV     R2,(SP)         ;SAVE POINTER TO NEXT CHAR
    601 002400  060166  000006                 ADD     R1,6(SP)        ;INCREMENT LINE POSITION
    602 002404  112742  000040          29$:   MOVB    #CHRSPA,-(R2)   ;MOVE IN THE SPACES
    603 002410                                 SOB     R1,29$
    604 002414                                 POP     <R0,R1>         ;RESTORE THE REGISTERS
    605                                 ;+
    606                                 ;  Process all else, e.g. Spaces, overstrikes, operators
    607                                 ;-
    608 002420  005266  000002          30$:   INC     2(SP)           ;ONE MORE CHAR IN THE LINE
    609 002424  005200                  31$:   INC     R0              ;INCR. POINTER
    610 002426  005301                         DEC     R1              ;DECREMENT COUNT
    611 002430  003434                         BLE     38$             ;TOO LONG, SO SO LONG
    612 002432  121027  000010                 CMPB    (R0),#CHR.BS    ;NEXT CHAR A BACKSPACE ?
    613 002436  001254                         BNE     21$             ;NO
    614
    615 002440  005200                         INC     R0              ;YES, SO REMOVE A CHAR
    616 002442                                 SOB     R1,31$          ;AND LOOP
    617 002446  000425                         BR      38$             ;NO MORE, FINISHED
    618
    619 002450  104541                  32$:   LNLNG                   ;BYE, BYE
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 55
FNEDIT          HANDLE FUNCTION EDITING


                                        ;+
                                        ;  Edit command buffer used up first, skip to end of function line buffer
                                        ;-
002452  005701                 33$:    TST     R1              ;ANY CHARACTERS LEFT ?
002454  003422                         BLE     38$             ;NO, DONT DO THIS
002456  016603  000002                 MOV     2(SP),R3        ;GET CURRENT POSITION
002462  121027  000010         34$:    CMPB    @R0,#CHR.BS     ;A BACKSPACE ?
002466  001002                         BNE     35$             ;NO, SO DON'T WORRY
002470  005303                         DEC     R3              ;DECREASE THE COLUMN # BY ONE
002472  000407                         BR      37$             ;AND GO
002474  121027  000015         35$:    CMPB    @R0,#CHR.CR     ;A <CR> ?
002500  001410                         BEQ     38$             ;YES, FINISHED
002502  121027  000040                 CMPB    @R0,#CHRSPA     ;DOES THIS CHAR PRINT ?
002506  002401                         BLT     37$             ;NO, SKIP IT
002510  005203                         INC     R3              ;MOVE OVER ONE COLUMN
002512  005200                 37$:    INC     R0              ;INCREMENT THE POINTER
002514                                 SOB     R1,34$          ;AND LOOP 'TIL FINISHED
002520  000402                         BR      36$
                                        ;+
                                        ;  Position cursor in proper place
                                        ;-
002522  016603  000002         38$:    MOV     2(SP),R3        ;Get current position
002526  060001                 36$:    ADD     R0,R1           ;POINT TO THE END OF THE STRING
002530  013704  005064'                 MOV     EDICNT,R4       ;GET THE START POSITION
002534  100410                         BMI     40$             ;START AT THE END
002536  160403                         SUB     R4,R3           ;HOW MANY BACKSPACES ?
002540  003406                         BLE     40$             ;TOO FEW
002542  112721  000010         39$:    MOVB    #CHR.BS,(R1)+   ;BACKING UP
002546  005316                         DEC     (SP)            ;DECREASE SPACE AVAILABLE
002550  003737                         BLE     32$             ;NONOE, ERROR
002552                                 SOB     R3,39$          ;LOOP A WHILE
002556  010137  000242'         40$:    MOV     R1,C.CPTR+CHANOO;Pointer past last data
002562  163701  000236'                 SUB     C.BUFF+CHANOO,R1;The number of real characters
002566  013737  000240' 000244'         MOV     C.BUFL+CHANOO,C.CCNT+CHANOO
                                                                ;Copy buffer size
002574  160137  000244'                 SUB     R1,C.CCNT+CHANOO;Compute space left in the buffer
002600  000137  000430'                 JMP     SAMLN1          ;AND CONTINUE
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 56
FNOPEN          OPEN FUNCTION FOR EDITING


                                        .SBTTL  FNOPEN          OPEN FUNCTION FOR EDITING

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       This routine is used to open a function for
                                        ;       editing; i.e., convert .INTEGER line numbers
                                        ;       to .FLOAT line numbers, and all such goodies.
                                        ;
                                        ; INPUTS:
                                        ;       R0      - Pointer to the Function Header
                                        ;
                                        ; ROUTINES CALLED:
                                        ;       RETCOR  - Returns the Label List Space to Memory
                                        ;--
002604                                          .PSECT  CODE

002604                                  FNOPEN::
002604  132760  000004  000003                  BITB    #F.LOC,R.BITS(R0) ;PREVIOUSLY PROTECTED ?
002612  001060                                  BNE     3$                ;YES, SO EDITING IS A NO-NO
002614  010037  005012'                         MOV     R0,FHEADR         ;SAVE POINTER TO FUNCTION HEADER BLOCK
002620  132760  000010  000003                  BITB    #F.OPN,R.BITS(R0) ;Already Open?
002626  001015                                  BNE     2$                ;Yes, skip this step
002630                                  1$:
002630                                          LDCIF   R.LNUM(R0),AC0    ;CONVERT TO FLOAT
002640                                          STF     AC0,R.LNUM(R0)    ;  AND REPLACE
002650  016000  000006                          MOV     R.NEXT(R0),R0     ;POINT TO NEXT ELEMENT
002654  020037  005012'                         CMP     R0,FHEADR         ;FINISHED?
002660  001363                                  BNE     1$                ;LOOP FOR ALL ELEMENTS
002662  016000  000004          2$:             MOV     R.PREV(R0),R0     ;Point to last frame
002666                                          LDF     R.LNUM(R0),AC0    ;Get the line number
002676                                          STF     AC0,LINE          ;OF COURSE
002706                                          LDF     #^F1.0,AC0        ;GET A 1.0
002716                                          STF     AC0,LININC        ;MAKE INITIAL LINE INCREMENT
002726  010037  005014'                         MOV     R0,FLINE          ;SAVE PTR TO LINE
002732  013700  005012'                         MOV     FHEADR,R0         ;Point to function header
002736  052737  000010  004764'                BIS     #F.FNDF,FLAGS     ;In function definition mode
002744  152760  000010  000003                  BISB    #F.OPN,R.BITS(R0) ;Mark function as Open
002752                                          RETURN

002754  104405                  3$:             DEFERR
```

```
                                .SBTTL  EOLCHK          CHECK FOR DEL AT LINE END

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'EOLCHK' checks for then end of the current input line.
                                ;       If the end of the line isn't found, a syntax error is
                                ;       signalled.  If a del is found, F.STOP Is set in the
                                ;       'FLAGS' word.
                                ;
                                ; INPUTS:
                                ;       R3 -    COUNT OF CHARS LEFT IN LINE
                                ;       R4 -    POINTER TO NEXT CHAR
                                ;--
002756                          .PSECT  CODE

002756                  EOLCHK: CALL    GETNB           ;GET NON-BLANK CHAR
002762  002406                  BLT     1$              ;NONE LEFT, ERROR
002764  122702  000015          CMPB    #CHR.CR,R2      ;EOL?
002770  001004                  BNE     2$              ;NO, TRY FOR A DEL
002772  042737  000004  004764' BIC     #F.STOP,FLAGS   ;NO DEL FOUND
003000                  1$:     RETURN                  ;ALL DONE

003002  122702  000107  2$:     CMPB    #CHRDEL,R2      ;DEL?
003006  001406                  BEQ     3$              ;YES, GOOD
003010  122702  000271          CMPB    #CHRPDL,R2      ;PROTECT DEL?
003014  001014                  BNE     5$              ;No
003016  052737  010000  004764' BIS     #F.PROT,FLAGS   ;YES, SET PROTECT BIT
003024  052737  000004  004764' 3$:     BIS     #F.STOP,FLAGS   ;YES, SET STOP BIT
003032                  4$:     CALL    GETNB           ;GET NON-BLANK CHAR
003036  002760                  BLT     1$              ;NONE LEFT, DONE
003040  122702  000015          CMPB    #CHR.CR,R2      ;EOL?
003044  001755                  BEQ     1$              ;YES, GOOD
003046  104407          5$:     SYNERR                  ;ELSE SYNTAX ERROR
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 59
EOLCHK          CHECK FOR DEL AT LINE END

      1         000001                              .END
```

FN MACRO M1113 08-FEB-80 12:20 PAGE 59-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073          CTRMAX  000172R     025 D.CTLM= 000000          EDIFLG  005062R     025
AC1    =%000001          CHRLPR= 000072          C.ALOC= 000162          D.DEAL= 000003          EDIT   = 000003
AC2    =%000002          CHRMNS= 000137          C.BLOK= 000064          D.DECD= 000002          EIS    = 000000
AC3    =%000003          CHRMUL= 000075          C.BLSZ= 000140          D.DIV = 000000          ELESIZ  005132R     025
AC4    =%000004          CHRNEG= 000100          C.BUFF= 177760          D.DMD = 000010          ENDCOR  000004R     025
AC5    =%000005          CHRPDL= 000271          C.BUFL= 177762          D.DQAD= 177777          EOF    = 104475
ALLINT  003274R     025  CHRPLU= 000055          C.CCNT= 177766          D.DROP= 000002          EOLCHK  002756R     020
ARYRET  003276R     025  CHRPWR= 000120          C.CPTR= 177764          D.ENCD= 000002          ERRMAX= 000166
ASCQUO= 000042           CHRQST= 000121          C.CZ   = 001000          D.EQ   = 000000          ERRNUM  004772R     025
A.ALEN= 000004           CHRQUO= 000113          C.DIG  = 000040          D.EXPD= 000002          ERRPTR  004770R     025
A.ARG  = 000040          CHRRBR= 000047          C.FASW= 020000          D.FGET= 000005          ERRO    004774R     025
A.ASB  = 000200          CHRRPR= 000042          C.FLEN= 000010          D.FMT  = 000004          EXEBUF  005054R     025
A.BITS= 000003           CHRSEM= 000074          C.FLSH= 004000          D.FPNT= 000005          EXEC   = ****** GX
A.CON  = 000020          CHRSMO= 000112          C.FRC  = 000001          D.FPUT= 000005          E$CT    005056R     025
A.DIMS= 000002           CHRSPA= 000040          C.IDEN= 000010          D.GE   = 000000          E$PT    005060R     025
A.DIM1= 000006           CHRTAB= ****** GX       C.NAME= 000102          D.GT   = 000000          FA.APD= 000100
A.NUM  = 000006          CHRUPA= 000131          C.NOBF= 000400          D.IBM  = 000002          FA.CRE= 000010
A.PTR  = 000000          CHR.BS= 000010          C.NUM  = 000020          D.IND  = 000002          FA.DLK= 001000
A.TMP  = 000010          CHR.CC= 000003          C.OP   = 000004          D.INNR= 000013          FA.ENB= 100000
BADDIR= 104463           CHR.CR= 000015          C.OPN  = 010000          D.INTR= 000006          FA.EXC= 002000
BADERR= 104437           CHR.CZ= 000032          C.RDLN= 177776          D.LE   = 000000          FA.EXT= 000004
BLZBUB  005116R     025  CHR.GT= 000046          C.REWT= 002000          D.LOG  = 000001          FA.NSP= 000100
BRSELF  005046R     025  CHR.LF= 000012          C.RND  = 000010          D.LT   = 000000          FA.NSR= 000200
CCNT    005072R     025  CHR.LT= 000043          C.SBIT= 177756          D.MAX  = 000000          FA.POS= 010000
CHANEX  003060R     025  CHR.SI= 000017          C.SPFX= 000100          D.MEM  = 000002          FA.RD  = 000001
CHAN00  000256R     025  CHR.SL= 000057          C.SPGL= 000200          D.MIN  = 000000          FA.RWD= 004000
CHAN01  000330R     025  CHR.SO= 000016          C.SPLX= 000001          D.MUL  = 000000          FA.SEQ= 040000
CHAN02  000512R     025  CHR.TB= 000011          C.SPRS= 000002          D.NE   = 000000          FA.SHR= 000040
CHAN03  000674R     025  CHR.UL= 000106          C.SRC  = 177774          D.NN   = 000000          FA.TMP= 000020
CHAN04  001056R     025  CHR..A= 000141          C.SRCP= 000024          D.NR   = 000000          FA.WCK= 020000
CHAN05  001240R     025  CHR..E= 000145          C.TT   = 000004          D.OR   = 000000          FA.WRT= 000002
CHAN06  001422R     025  CHR..S= 000163          C.WIDC= 177770          D.PLUS= 000000          FD.BLK= 000010
CHAN07  001604R     025  CHR..T= 000164          C.WIDR= 177772          D.PSUB= 000006          FD.CCL= 000002
CHAN08  001766R     025  CHR..U= 000165          DEBUG  = 000000          D.PWR  = 000001          FD.COM= 020000
CHAN09  002150R     025  CHR..Z= 000172          DEFERR= 104405          D.QQAD= 000000          FD.CR  = 000002
CHAN10  002332R     025  CHR..0= 000060          DEVIL   005114R     025 D.QUAD= 000000          FD.DIR= 000010
CHAN11  002514R     025  CHR..1= 000061          DEVNTF= 104470          D.RES  = 000000          FD.FTN= 000001
CHAN12  002676R     025  CH.AND= 000001          DIGITS  000006R     025 D.RHO  = 000000          FD.F11= 040000
CHAN13  000022R     023  CKFNSI= ****** GX       DIMAX  = 000020          D.ROT  = 000003          FD.INS= 000010
CHAN14  000204R     023  CMDBUF   003250R    025 DIMBLK  003112R     025 D.SCON= 000006          FD.ISP= 002000
CHNOPN= 104471           CMDMAX= 000010          DIMPTR  003114R     025 D.SDIF= 000006          FD.MNT= 100000
CHRAND= 000051           CMP.F  = ****** GX      DIMTYP  003116R     025 D.SUB  = 000000          FD.OSP= 004000
CHRBEL= 000007           CNTSIZ  005130R     025 DIRFUL= 104466          D.SUBS= 000006          FD.PLC= 000004
CHRBKR= 000133           COMBIN  005076R     025 DOMERR= 104417          D.TAKE= 000002          FD.PRN= 000004
CHRBOL= 000200           COMCNT  005126R     025 DPTERR= 104416          D.TRAN= 000004          FD.PSE= 010000
CHRBOX= 000114           COMERR= 104425          DYDCNT  003224R     025 D.UNON= 000006          FD.RAH= 000001
CHRCMA= 000054           COMPTR  005122R     025 D.AND  = 000000          D.$MAX= 000000          FD.RAN= 000002
CHRCOL= 000076           COMRTN  003156R     025 D.BXCT= 000007          D.$MIN= 000000          FD.REC= 000001
CHRDEL= 000107           COMSKP  005124R     025 D.BXIO= 000007          D.$MUL= 000000          FD.RWM= 000001
CHRDIV= 000053           CORPTR  005120R     025 D.BXNL= 000007          D.$PLU= 000000          FD.SDI= 000020
CHRDLR= 000176           CS.ASS= 000200          D.BXPP= 000007          D.$RES= 000000          FD.SQD= 000040
CHRDLT= 000110           CS.BOO= 000010          D.BXPW= 000007          D.$SUB= 000000          FD.TTY= 000004
CHRDOT= 000056           CS.FLO= 000020          D.BXRL= 000007          D.1COM= 000001          FD.WBH= 000002
CHREP  = 000105          CS.IDN= 000040          D.CIRC= 000001          D.1EXP= 000002          FF.CHR= 000005
CHRERR= 104431           CS.ONE= 000100          D.COMB= 000001          D.1ROT= 000003          FF.NV  = 000003
CHRFIL= 000177           CTEPS   000152R     025 D.COMP= 000001          EDICHR  005063R     025 FF.POE= 000002
CHRFRO= 000135           CTMAX   000212R     025 D.CONT= 000006          EDICNT  005064R     025 FF.RWD= 000001
```

SYMBOL TABLE

```
FF.RWF= 000006          F.BKP1= 000051          F.SEQN= 000100          LFTOP   005066R 025 M.GETI= 000006
FF.SPC= 000004          F.BKST= 000024          F.SPDV= 000072          LINC    005102R 025 M.IBM = 000003
FHEADR  005012R 025     F.BKVB= 000064          F.SPUN= 000074          LINE    005016R 025 M.IDN = 000000
FHRET   005040R 025     F.BR  = 000020          F.STBK= 000036          LININC  005026R 025 M.INSP= 000006
FILNTF= 104467          F.CHR = 000075          F.STK = 000022          LINKIL= ****** GX       M.IOTA= 000000
FIS   = 000000          F.CNTG= 000034          F.STOP= 00C004          LISIZE= 000020          M.LIT = 000205
FISCON= 046000          F.CODE= 000010          F.STP = 000001          LITOP   005072R 025 M.LN  = 000001
FIXLIN= ****** GX       F.DFNB= 000046          F.SUSP= 000002          LNLNG = 104541          M.MMD = 000010
FIXPTR  003074R 025     F.DSPT= 000044          F.TRC = 000002          LOCLCT  005036R 025 M.NEG = 000000
FLAGS   004764R 025     F.DVNM= 000134          F.UNIT= 000136          LOWCOR  000002R 025 M.NOT = 000000
FLCBUF  003250R 025     F.EFBK= 000010          F.URBD= 000020          LSTERP  004766R 025 M.NUB = 000006
FLINE   005014R 025     F.EFN = 000050          F.VAL = 177776          LWIDTH  000070R 025 M.NUM = 000204
FLNLST= ****** GX       F.END = 002000          F.VBN = 000064          L.BITS= 000003          M.OUTR= 000012
FLOEXP  003270R 025     F.EOBB= 000032          F.VBSZ= 000060          L.CNT = 000002          M.PITM= 000000
FLONEG  003272R 025     F.ERR = 000052          GETLIN= ****** GX       L.PRNT= 000001          M.PRDS= 000006
FLOUT = ****** GX       F.FACC= 000043          GETNB = ****** GX       L.PTR = 000000          M.QQAD= 000000
FLSWTB  000222R 025     F.FFBY= 000014          GETNB2= ****** GX       L.PTR1= 000004          M.QUAD= 000000
FLTLEN= 000010          F.FLAG= 000002          GNOME   003102R 025     L.SUBS= 000002          M.RAND= 000012
FN      000000RG 020    F.FLB = 000100          HASHC   005136R 025     MACHIN= 000106          M.RAVL= 000000
FND     000262RG 020    F.FNAM= 000110          HASHTB  005140R 025     MAXCHN= 000014          M.RCIP= 000000
FNEDDL  001140R 020     F.FNB = 000102          HSHMSK= 177601          MONBLK  003120R 025 M.RDCT= 000011
FNEDIT  000642R 020     F.FNDF= 000010          HSHSIZ= 000200          MONCVT  003132R 025 M.REVS= 000003
FNEDTX  001074R 020     F.FTYP= 000116          ICHAN   005050R 025     MONLEN  003130R 025 M.ROLL= 000000
FNEDT1  001306R 020     F.FUN = 000010          ILFNAM= 104464          MONPTR  003122R 025 M.SATA= 000001
FNEND = ****** GX       F.FVER= 000120          ILLCHN= 104540          MONSFT  003126R 025 M.SCAN= 000011
FNG     000300R 020     F.HEAD= 000006          ILLSW = 104565          MONTYP  003124R 025 M.SHAP= 000000
FNHEAD= ****** GX       F.HIBK= 000004          INDERR= 104410          MONVAL  003134R 025 M.SIGN= 000000
FNINSI  005044R 025     F.LBL = 000020          INNTMP  003250R 025     M.ABS = 000000          M.SUBR= 000001
FNLINE= ****** GX       F.LINE= 000012          INNTM2  005016R 025     M.BOL = 000200          M.SYM = 000203
FNLIST= ****** GX       F.LOC = 000004          INPTR   000064R 025     M.BXAV= 000007          M.SYN = 000201
FNOPEN  002604RG 020    F.LSTC= 000004          INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FNRDF   000266R 020     F.LUN = 000042          IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FO.APD= 000106          F.MBC1= 000055          I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FO.MFY= 000002          F.MBFG= 000056          I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.RD = 000001          F.MON = 000002          I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.UPD= 000006          F.NIL = 000001          I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.WRT= 000016          F.NRBD= 000024          I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FPMODE  000042R 025     F.NREC= 000030          I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FPP   = 000000          F.OPN = 000010          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FPPLEN= 000004          F.OVBS= 000030          LABLPT  005042R 025     M.BXRL= 000007          NAMBUF  003240R 025
FREBLK  007400R 025     F.PREV= 000000          LARGBL  003162R 025     M.BXTS= 000007          NB.DEV= 000200
FUZZ    000142R 025     F.PRNT= 100000          LARGCV  003176R 025     M.BXWA= 000007          NB.DIR= 000100
FXPTR1  003076R 025     F.PROT= 010000          LARGFL  003160R 025     M.CEI = 177777          NB.NAM= 000004
F$EVAL= 000100          F.RACC= 000016          LARGLE  003174R 025     M.CHR = 000202          NB.SD1= 000400
F$EVI = 000200          F.RATT= 000001          LARGLN  003166R 025     M.CMT = 000206          NB.SD2= 001000
F$EXEC= 000400          F.RCNM= 000034          LARGPT  003164R 025     M.DOUT= 000006          NB.SNM= 000040
F$FUNC= 001000          F.RCTL= 000017          LARGSF  003172R 025     M.DQAD= 000006          NB.STP= 000020
F.ACTL= 000076          F.READ= 004000          LARGTY  003170R 025     M.ENCD= 000004          NB.SVR= 000010
F.ALOC= 000040          F.REDF= 000040          LARGVA  003200R 025     M.EXE = 000000          NB.TYP= 000002
F.BBFS= 000062          F.RES = 000040          LBINC   005100R 025     M.EXP = 000001          NB.VER= 000001
F.BDB = 000070          F.RET = 177772          LBUFF   003316R 025     M.FACT= 000001          ND.DEV= 000200
F.BGBC= 000057          F.RSIZ= 000002          LBUFLN= 000621          M.FIL = 000177          ND.EXT= 000002
F.BITS= 177774          F.RTYP= 000000          LCOUNT  005106R 025     M.FLR = 177777          ND.NAM= 000004
F.BKDN= 000026          F.R2  = 000014          LENERR= 104412          M.FMT = 000004          ND.PPN= 000100
F.BKDS= 000020          F.R3  = 000016          LENTAB  000054R 025     M.GDUP= 000003          ND.$EX= 000000
F.BKEF= 000050          F.R4  = 000020          LEXPTR  003100R 025     M.GDWN= 000003          ND.$E$= 000020
```

SYMBOL TABLE

```
ND.$NM= 000000          OPTMP2  005072R     025 ROVER   003312R     025 SYMWRN= 000054          T.MON = 000010
ND.$NS= 000040          OPTMP3  005074R     025 RSTS  = 000000          SYNERR= 104407          T.NIL = 000004
ND.$PN= 001400          OPTMP4  005076R     025 RSX   = 000115          SYSERR= 104402          T.NUM = 002000
NEG$GN  005063R     025 OPTMP5  005100R     025 RTOP    005070R     025 SYSTAT  000000R     024 T.OPFN= 000001
NEWLIN= ****** GX       OPTMP6  005102R     025 RT11  = 000000          SY.BAD= 000001          T.OTHR= 040000
NEXTER  000640R     020 OPTMP7  005104R     025 R.BITS= 000003          SY.BRK= 100000          T.SOP = 000002
NEXTLN  000320RG    020 OPTMP8  005106R     025 R.FIX = 000001          SY.BS = 000010          T.SPOP= 000040
NINPT1  003302R     025 OPTMP9  005110R     025 R.LCT = 000002          SY.CLR= 000040          T.TMP = 000200
NINPT2  003300R     025 OPTM10  005112R     025 R.LN  = 000002          SY.ERP= 000020          T.VAL = 100000
NINSTK  003304R     025 OPTM14  005026R     025 R.LNUM= 000010          SY.NOU= 000100          T.VAR = 001000
NINTMP  003310R     025 ORIGIN  000030R     025 R.NEXT= 000006          SY.OPN= 000002          UPPER = 177400
NINTYP  003306R     025 OUTCHR= ****** GX       R.OFF = 000000          SY.PMT= 001000          UTNAME  000344R     023
NOTAVL= 104472          OUTPTR  000066R     025 R.PREV= 000004          SY.SRT= 000400          VALERR= 104413
NOTOPN= 104473          OVROPR= 000000          R.PTR = 000000          SY.SVC= 000004          VARBLE  007400R     025
NPRT1   004762R     025 PATCH   007344R     025 R.SEQ = 000003          SY.SVL= 000200          VARBLS  000000R     025
NULL    007364R     025 PRECSN  000052R     025 R.VAR = 000002          S.AB  = 177776          VERNUM  000000R     025
NUMLER= ****** GX       PRINTM= ****** GX       R6    =%000006          S.ASB = 100000          VRSION= 000002
NUMCHN= 000020          RANDAC  003064R     025 R7    =%000007          S.BFHD= 000020          WIDTH   000072R     025
NUMDFL  003230R     025 RANDY   003062R     025 SAMELN  000350RG    020 S.BITS= 000002          WROUND  000020R     025
NUMEFL  003233R     025 RANERR= 104411          SAMLN1  000430RG    020 S.CHR1= 000003          WSDATE  000136R     025
NUMESN  003236R     025 RARGBL  003204R     025 SAMLN2  000462RG    020 S.FATT= 000016          WSFULL= 104403
NUMFLG  003230R     025 RARGCV  003220R     025 SI    =%000005          S.FDB = 000140          WSNAME  000074R     025
NUMLEN  000062R     025 RARGFL  003202R     025 SIDAMN= 104433          S.FLB = 040000          WSTIME  000132R     025
NUMNEG  003232R     025 RARGLE  003216R     025 SIFRAM  003314R     025 S.FNAM= 000006          $ADDF = ****** GX
NUMQFL  003234R     025 RARGLN  003210R     025 SISTAK  006350R     025 S.FNB = 000036          $CFCC = ****** GX
NUMRIG  003231R     025 RARGPT  003206R     025 SITOP   006346R     025 S.FNBW= 000017          $CMPF = ****** GX
NUMWID  003226R     025 RARGSF  003214R     025 SITP    007144R     025 S.FNTY= 000004          $DIVF = ****** GX
NUMZFL  003235R     025 RARGTY  003212R     025 SIWARN= 000200          S.FTYP= 000002          $FIS  = ****** G
NYIERR= 104432          RARGVA  003222R     025 SIZTAB  000044R     025 S.LEN = 000002          $FPP  = ****** G
N.BLSZ= 000036          RBASE   005074R     025 SI.BIT= 177774          S.LINK= 177774          $FPPLN= ****** G
N.DESC= 000020          RBRTXT= ****** GX       SI.LEN= 177772          S.NFEN= 000020          $LDCIF= ****** GX
N.DEV = 000032          RCOUNR  005112R     025 SI.RET= 177772          S.TYPE= 140000          $LDF  = ****** GX
N.DEVN= 000034          RCOUNT  005110R     025 SI.VAL= 177776          S.VAL = 000000          $MACHN= ****** G
N.DID = 000024          REDCNT  005070R     025 SKPSIZ  005134R     025 TKDIMS  005072R     025 $RSTS = ****** G
N.DVNM= 000032          REDIGN  005066R     025 SPHOLD  000004R     024 TKFRAM  005066R     025 $RSX  = ****** G
N.EXT = 000014          REDOFF  005072R     025 SPINIT  000002R     024 TKLEFT  005070R     025 $RT11 = ****** G
N.FID = 000000          REENTR  003060R     025 STKINC= 000774          TMODE   000403R     023 $STCFI= ****** GX
N.FNAM= 000006          RESBIT  003142R     025 STOP  = 104420          TMPTAG= 000166          $STF  = ****** GX
N.FTYP= 000014          RESCVT  003152R     025 SUBCNT  003104R     025 TTINIT= ****** GX       $SUBF = ****** GX
N.FVER= 000016          RESERR= 104440          SUBDCT  003106R     025 TTYBUF  004140R     025 $TSTF = ****** GX
N.NAME= 000006          RESLEN  003150R     025 SUBPTR  003110R     025 TTYLEN= 000622          .BOOLE= 000002
N.NEXT= 000022          RESPTR  003140R     025 SYMBOL= ****** GX       TURKEY= 104436          .CHARA= 000000
N.PPN = 000024          RESSFT  003146R     025 SYMB1   005344R     025 TYPE  = 000166          .DEF. = 000002
N.STAT= 000020          RESTYP  003144R     025 SYMF    005340R     025 T.BOL = 020000          .FLOAT= 000006
N.UNIT= 000034          RESULT  003136R     025 SYMPT1  005346R     025 T.CON = 004000          .FSRCB= ****** G
OBUFLN= 001000          RESVAL  003154R     025 SYMTAB  005352R     025 T.DIM = 000100          .INTEG= 000004
OCHAN   005052R     025 RINC    005104R     025 SYMTBF= 104404          T.FUN = 010000          .MOLUN  000404R     023
OMODE   000402R     023 RLTBAS  005066R     025 SYMTOP  005350R     025 T.LIST= 000400          ...GBL= 000000
OPTMP0  005066R     025 ROTINC  005070R     025 SYMTP   006272R     025 T.LIT = 000020          ...TPC= 000020
OPTMP1  005070R     025 ROUND   000010R     025
```

```
. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
```

```
FN      MACRO M1113  08-FEB-80 12:20  PAGE 59-4
SYMBOL TABLE

MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    003050      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13551 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:04:10
FN,DM1:[305,301]FN/-SP=APLRX1,APLDEF,FN,END
```

FILEID**FNEND

```
FFFFFFFFFF    NN       NN    EEEEEEEEEE    NN       NN    DDDDDDDD
FFFFFFFFFF    NN       NN    EEEEEEEEEE    NN       NN    DDDDDDDD
FF            NN       NN    EE            NN       NN    DD      DD
FF            NN       NN    EE            NN       NN    DD      DD
FF            NNNN     NN    EE            NNNN     NN    DD      DD
FF            NNNN     NN    EE            NNNN     NN    DD      DD
FFFFFFFF      NN  NN   NN    EEEEEEEE      NN  NN   NN    DD      DD
FFFFFFFF      NN  NN   NN    EEEEEEEE      NN  NN   NN    DD      DD
FF            NN    NNNN    EE            NN    NNNN    DD      DD
FF            NN    NNNN    EE            NN    NNNN    DD      DD
FF            NN      NN    EE            NN      NN    DD      DD      ....
FF            NN      NN    EE            NN      NN    DD      DD      ....
FF            NN      NN    EEEEEEEEEE    NN      NN    DDDDDDDD        ....
FF            NN      NN    EEEEEEEEEE    NN      NN    DDDDDDDD        ....


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 12:24
TABLE OF CONTENTS


```
     2-    1     APLDEF
     2-    8     A P L
     2-    9
     2-   10     A PROGRAMMING LANGUAGE
     2-   11
     2-   12     STEVE POULSEN & DAVE VANN
     2-   13     11-Jul-76
     2-   14     ROBERT GOTTLIEB & PETER HAYNES
     2-   15     2-Feb-79
     2-   16
     2-   17     V2
     2-   18
     2-   19
     2-   20     COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21     DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24     THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25     COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26     COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27     BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28     USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29     TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30     DEC.
     2-   31
     2-   32     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33     AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34     CORPORATION.
     2-   35
     2-   36     DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37     SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46     DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139     DEFINITIONS      OPERATOR OVERLAYS
     5-  268     DEFINITIONS      REGISTERS
     6-  331     DEFINITIONS      MEMORY ORGANIZATION
     7-  377     DEFINITIONS      ORDERING PSECTS
     8-  426     DEFINITIONS      SI PARSE FRAMES
     9-  474     DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515     DEFINITIONS      SI EXECUTION FRAMES
    11-  580     DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607     DEFINITIONS      CHRTAB BITS
    12-  624     DEFINITIONS      SYSTEM STATUS BITS
    13-  640     DEFINITIONS      F-FRAMES
    14-  724     DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807     DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869     DEFINITIONS      LIST STORAGE BLOCK
    18-  911     DEFINITIONS      INDEXING STACK FRAME
    19-  940     DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969     DEFINITIONS      CHANNEL BLOCK
    22- 1085     DEFINITIONS              RSX DATA
    23- 1110     DEFINITIONS      NAME BLOCKS
    33- 1612     DEFINITIONS      MACROS
    35- 1905     DEFINITIONS      APL CHARACTERS
    36- 1967     DEFINITIONS      ERRORS
    44-    4     FNEND            CLOSE FUNCTION DEFINITION
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  FNEND
                                        .IDENT  \V02.03\

                                        .SBTTL  FNEND           CLOSE FUNCTION DEFINITION

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       'FNEND' is used to close an open function.  A pass
                                        ;       is made through all of the function's lines so that
                                        ;       the line numbers can be resequenced by 1 and so
                                        ;       that lines with labels can be identified.  Pointers
                                        ;       to lines with labels are pushed on the SP stack so
                                        ;       that a list block containing the pointers can be
                                        ;       attached to the function header.  This is needed so
                                        ;       that when the function is executed the labels will
                                        ;       be treated as local variables.  Finally, the function
                                        ;       is marked as closed and function definition mode
                                        ;       is exited.
                                        ;
                                        ; REGISTER ASSIGNMENTS:
                                        ;       R0 -    SCRATCH
                                        ;       R1 -    PTR TO CURRENT LINE
                                        ;       R2 -    COUNT OF LABELS FOUND
                                        ;       R3 -    INTEGER LINE #
                                        ;       R4 -    PTR TO FUNCTION'S HEAD
                                        ;       AC0 -   CURRENT LINE #
                                        ;       AC1 -   1.0
                                        ;--
000000                                  .PSECT  CODE

000000  013704  005012'         FNEND:: MOV     FHEADR,R4       ;POINT TO FN HEADER
000004  010400                          MOV     R4,R0           ;COPY POINTER
000006  062700  000010                  ADD     #R.LNUM,R0      ;POINT TO LINE NUMBER
000012  063700  000062'                 ADD     NUMLEN,R0       ;POINT TO LABEL LIST BLOCK PTR
000016  010037  003136'                 MOV     R0,RESULT       ;SAVE LABEL POINTER
000022  011000                          MOV     @R0,R0          ;ANY OLD LABEL LIST?
000024  001404                          BEQ     1$              ;NOPE
000026                                  CALL    RETCOR          ;YES, KILL IT
000032  005077  003136'                 CLR     @RESULT         ;ZERO WORD IN FN HEADER
000036  005003                  1$:     CLR     R3              ;CLEAR INTEGER LINE #
000040  005002                          CLR     R2              ;CLEAR COUNT OF LABELS FOUND
000042  010401                          MOV     R4,R1           ;POINT TO FIRST LINE
000044  000407                          BR      3$              ;AND START PROCESSING LINES

000046  010361  000010          2$:     MOV     R3,R.LNUM(R1)   ;STORE NEW LINE NUMBER IN LINE BLOCK
000052  105761  000002                  TSTB    R.LCT(R1)       ;LINE HAVE LABEL?
000056  001402                          BEQ     3$              ;NOPE
000060  005202                          INC     R2              ;YES, COUNT IT
000062                                  PUSH    R1              ;AND SAVE POINTER TO LINE

000064  005203                  3$:     INC     R3              ;BUMP INTEGER LINE #
000066  016101  000006                  MOV     R.NEXT(R1),R1   ;POINT TO NEXT LINE
000072  020104                          CMP     R1,R4           ;DONE?
000074  001364                          BNE     2$              ;NO, PROCESS NEXT LINE
                                        ;+
                                        ; FUNCTION IS RESEQUENCED AND LABELLED LINES HAVE BEEN
                                        ; LOCATED.  COPY POINTERS INTO LIST BLOCK AND FINISH UP.
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 44-1
FNEND           CLOSE FUNCTION DEFINITION

     58                                     ;-
     59 000076  005702                      TST     R2              ;ANY LABELS FOUND?
     60 000100  001421                      BEQ     5$              ;NO
     61 000102  010200                      MOV     R2,R0           ;COPY LABEL COUNT
     62 000104  006300                      ASL     R0              ;GET WORD COUNT
     63 000106  062700  000004              ADD     #L.PTR1,R0      ;ADD OVERHEAD
     64 000112                              CALL    GETCOR          ;GET A BLOCK OF STORAGE FOR IT
     65 000116  010077  003136'             MOV     R0,@RESULT      ;SET PTR TO LIST BLOCK IN FN HEADER
     66 000122  010001                      MOV     R0,R1           ;COPY POINTER
     67 000124  062701  000004              ADD     #L.PTR1,R1      ;GET OFFSET TO FIRST PTR
     68 000130  010110                      MOV     R1,@R0          ;SAVE POINTER TO FIRST PTR
     69 000132  010260  000002              MOV     R2,L.CNT(R0)    ;FILL IN COUNT/CLEAR BITS
     70 000136  012621                  4$: MOV     (SP)+,(R1)+     ;COPY POINTER INTO BLOCK
     71 000140                              SOB     R2,4$           ;FILL IN ALL LABEL PTRS
     72
     73 000144  142764  000010  000003  5$: BICB    #F.OPN,R.BITS(R4);FUNCTION IS NO LONGER OPEN
     74 000152  032737  010000  004764'      BIT     #F.PROT,FLAGS   ;WAS IT PROTECTED?
     75 000160  001403                      BEQ     6$              ;NO
     76 000162  152764  000004  000003      BISB    #F.LOC,R.BITS(R4);ITS LOCKED
     77 000170  042737  010014  004764' 6$: BIC     #F.FNDF+F.STOP+F.PROT,FLAGS;NO LONGER IN FN DEF MODE
     78 000176                              RETURN                  ;   and return
     79
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 46
FNEND           CLOSE FUNCTION DEFINITION

      1         000001                          .END
```

FNEND MACRO M1113 08-FEB-80 12:24 PAGE 46-1
SYMBOL TABLE

```
AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R  025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R  025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R  025
AC5    =%000005          CHRPDL= 000271    C.CZ  = 001000    D.EQ  = 000000    ERRPTR  004770R  025
ALLINT  003274R  025     CHRPLU= 000055    C.DIG = 000040    D.EXPD= 000002    ERRO    004774R  025
ARYRET  003276R  025     CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R  025
ASCQUO= 000042           CHRQST= 000121    C.FLEN= 000010    D.FMT = 000004    E$CT    005056R  025
A.ALEN= 000004           CHRQUO= 000113    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R  025
A.ARG = 000040           CHRRBR= 000047    C.FRC = 000001    D.FPUT= 000005    FA.APD= 000100
A.ASB = 000200           CHRRPR= 000042    C.IDEN= 000010    D.GE  = 000000    FA.CRE= 000010
A.BITS= 000003           CHRSEM= 000074    C.NAME= 000102    D.GT  = 000000    FA.DLK= 001000
A.CON = 000020           CHRSMO= 000112    C.NOBF= 000400    D.IBM = 000002    FA.ENB= 100000
A.DIMS= 000002           CHRSPA= 000040    C.NUM = 000020    D.IND = 000002    FA.EXC= 002000
A.DIM1= 000006           CHRUPA= 000131    C.OP  = 000004    D.INNR= 000013    FA.EXT= 000004
A.NUM = 000006           CHR.BS= 000010    C.OPN = 010000    D.INTR= 000006    FA.NSP= 000100
A.PTR = 000000           CHR.CC= 000003    C.RDLN= 177776    D.LE  = 000000    FA.NSR= 000200
A.TMP = 000010           CHR.CR= 000015    C.REWT= 002000    D.LOG = 000001    FA.POS= 010000
BADDIR= 104463           CHR.CZ= 000032    C.RND = 000010    D.LT  = 000000    FA.RD = 000001
BADERR= 104437           CHR.GT= 000046    C.SBIT= 177756    D.MAX = 000000    FA.RWD= 004000
BLZBUB  005116R  025     CHR.LF= 000012    C.SPFX= 000100    D.MEM = 000002    FA.SEQ= 040000
BRSELF  005046R  025     CHR.LT= 000043    C.SPGL= 000200    D.MIN = 000000    FA.SHR= 000040
CCNT    005072R  025     CHR.SI= 000017    C.SPLX= 000001    D.MUL = 000000    FA.TMP= 000020
CHANEX  003060R  025     CHR.SL= 000057    C.SPRS= 000002    D.NE  = 000000    FA.WCK= 020000
CHAN00  000256R  025     CHR.SO= 000016    C.SRC = 177774    D.NN  = 000000    FA.WRT= 000002
CHAN01  000330R  025     CHR.TB= 000011    C.SRCP= 000024    D.NR  = 000000    FD.BLK= 000010
CHAN02  000512R  025     CHR.UL= 000106    C.TT  = 000004    D.OR  = 000000    FD.CCL= 000002
CHAN03  000674R  025     CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN04  001056R  025     CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FD.CR = 000002
CHAN05  001240R  025     CHR..S= 000163    DEBUG = 000000    D.PWR = 000001    FD.DIR= 000010
CHAN06  001422R  025     CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN07  001604R  025     CHR..U= 000165    DEVIL   005114R  025 D.QUAD= 000000  FD.F11= 040000
CHAN08  001766R  025     CHR..Z= 000172    DEVNTF= 104470    D.RES = 000000    FD.INS= 000010
CHAN09  002150R  025     CHR..0= 000060    DIGITS  000006R  025 D.RHO = 000000  FD.ISP= 002000
CHAN10  002332R  025     CHR..1= 000061    DIMAX = 000020    D.ROT = 000003    FD.MNT= 100000
CHAN11  002514R  025     CH.AND= 000001    DIMBLK  003112R  025 D.SCON= 000006  FD.OSP= 004000
CHAN12  002676R  025     CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006  FD.PLC= 000004
CHAN13  000022R  023     CMDMAX= 000010    DIMTYP  003116R  025 D.SUB = 000000  FD.PRN= 000004
CHAN14  000204R  023     CNTSIZ  005130R  025 DIRFUL= 104466    D.SUBS= 000006    FD.PSE= 010000
CHNOPN= 104471           COMBIN  005076R  025 DOMERR= 104417    D.TAKE= 000002    FD.RAH= 000001
CHRAND= 000051           COMCNT  005126R  025 DPTERR= 104416    D.TRAN= 000004    FD.RAN= 000002
CHRBEL= 000007           COMERR= 104425    DYDCNT  003224R  025 D.UNON= 000006  FD.REC= 000001
CHRBKR= 000133           COMPTR  005122R  025 D.AND = 000000    D.$MAX= 000000    FD.RWM= 000001
CHRBOL= 000200           COMRTN  003156R  025 D.BXCT= 000007    D.$MIN= 000000    FD.SDI= 000020
CHRBOX= 000114           COMSKP  005124R  025 D.BXIO= 000007    D.$MUL= 000000    FD.SQD= 000040
CHRCMA= 000054           CORPTR  005120R  025 D.BXNL= 000007    D.$PLU= 000000    FD.TTY= 000004
CHRCOL= 000076           CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000    FD.WBH= 000002
CHRDEL= 000107           CS.BOO= 000010    D.BXPW= 000007    D.$SUB= 000000    FF.CHR= 000005
CHRDIV= 000053           CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001    FF.NV = 000003
CHRDLR= 000176           CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002    FF.POE= 000002
CHRDLT= 000110           CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003    FF.RWD= 000001
CHRDOT= 000056           CTEPS   000152R  025 D.COMP= 000001    EDICHR  005063R  025 FF.RWF= 000006
CHREP = 000105           CTMAX   000212R  025 D.CONT= 000006    EDICNT  005064R  025 FF.SPC= 000004
CHRERR= 104431           CTRMAX  000172R  025 D.CTLM= 000000    EDIFLG  005062R  025 FHEADR  005012R  025
CHRFIL= 000177           C.ALOC= 000162    D.DEAL= 000003    EDIT   = 000003       FHRET   005040R  025
CHRFRO= 000135           C.BLOK= 000064    D.DECD= 000002    EIS    = 000000       FILNTF= 104467
```

SYMBOL TABLE

```
FIS    = 000000          F.FNB  = 000102          ICHAN   005050R  025 MONPTR  003122R  025 M.SATA= 000001
FISCON= 046000           F.FNDF= 000010          ILFNAM= 104464        MONSFT  003126R  025 M.SCAN= 000011
FIXPTR  003074R  025 F.FTYP= 000116          ILLCHN= 104540        MONTYP  003124R  025 M.SHAP= 000000
FLAGS   004764R  025 F.FUN  = 000010          ILLSW  = 104565        MONVAL  003134R  025 M.SIGN= 000000
FLCBUF  003250R  025 F.FVER= 000120          INDERR= 104410        M.ABS  = 000000          M.SUBR= 000001
FLINE   005014R  025 F.HEAD= 000006          INNTMP  003250R  025 M.BOL  = 000200          M.SYM  = 000203
FLOEXP  003270R  025 F.HIBK= 000004          INNTM2  005016R  025 M.BXAV= 000007          M.SYN  = 000201
FLONEG  003272R  025 F.LBL  = 000020          INPTR   000064R  025 M.BXCR= 000007          M.TRAN= 000012
FLSWTB  000222R  025 F.LINE= 000012          INSNAM= 104462        M.BXCT= 000007          M.$ABS= 000000
FLTLEN= 000010           F.LOC  = 000004          IOERR  = 104477        M.BXEX= 000007          M.$CEI= 000000
FNEND   000000RG  020 F.LSTC= 000004          I.CNT  = 177776        M.BXFX= 000007          M.$FLR= 000000
FNINSI  005044R  025 F.LUN  = 000042          I.DIM  = 177772        M.BXIO= 000007          M.$IDN= 000000
FORIGN  000032R  025 F.MBCT= 000054          I.LEN  = 000012        M.BXLC= 000003          M.$NEG= 000000
FO.APD= 000106           F.MBC1= 000055          I.OERR= 000062        M.BXNC= 000007          M.$SIG= 000000
FO.MFY= 000002           F.MBFG= 000056          I.OLD  = 177770        M.BXNL= 000007          M.1RDT= 000011
FO.RD  = 000001          F.MON  = 000002          I.PTR  = 177774        M.BXPP= 000007          M.1REV= 000003
FO.UPD= 000006           F.NIL  = 000001          I.STR  = 177766        M.BXPW= 000007          M.1SCN= 000011
FO.WRT= 000016           F.NRBD= 000024          LABERR= 104406        M.BXRL= 000007          NAMBUF  003240R  025
FPMODE  000042R  025 F.NREC= 000030          LABLPT  005042R  025 M.BXTS= 000007          NB.DEV= 000200
FPP    = 000000          F.OPN  = 000010          LARGBL  003162R  025 M.BXWA= 000007          NB.DIR= 000100
FPPLEN= 000004           F.OVBS= 000030          LARGCV  003176R  025 M.CEI  = 177777          NB.NAM= 000004
FREBLK  007400R  025 F.PREV= 000000          LARGFL  003160R  025 M.CHR  = 000202          NB.SD1= 000400
FUZZ    000142R  025 F.PRNT= 100000          LARGLE  003174R  025 M.CMT  = 000206          NB.SD2= 001000
FXPTR1  003076R  025 F.PROT= 010000          LARGLN  003166R  025 M.DOUT= 000006          NB.SNM= 000040
F$EVAL= 000100           F.RACC= 000016          LARGPT  003164R  025 M.DQAD= 000006          NB.STP= 000020
F$EVI  = 000200          F.RATT= 000001          LARGSF  003172R  025 M.ENCD= 000004          NB.SVR= 000010
F$EXEC= 000400           F.RCNM= 000034          LARGTY  003170R  025 M.EXE  = 000000          NB.TYP= 000002
F$FUNC= 001000           F.RCTL= 000017          LARGVA  003200R  025 M.EXP  = 000001          NB.VER= 000001
F.ACTL= 000076           F.READ= 004000          LBINC   005100R  025 M.FACT= 000001          ND.DEV= 000200
F.ALOC= 000040           F.REDF= 000040          LBUFF   003316R  025 M.FIL  = 000177          ND.EXT= 000002
F.BBFS= 000062           F.RES  = 000040          LBUFLN= 000621        M.FLR  = 177777          ND.NAM= 000004
F.BDB  = 000070          F.RET  = 177772          LCOUNT  005106R  025 M.FMT  = 000004          ND.PPN= 000100
F.BGBC= 000057           F.RSIZ= 000002          LENERR= 104412        M.GDUP= 000003          ND.$EX= 000000
F.BITS= 177774           F.RTYP= 000000          LENTAB  000054R  025 M.GDWN= 000003          ND.$E$= 000020
F.BKDN= 000026           F.R2   = 000014          LEXPTR  003100R  025 M.GETI= 000006          ND.$NM= 000000
F.BKDS= 000020           F.R3   = 000016          LFTOP   005066R  025 M.IBM  = 000003          ND.$N$= 000040
F.BKEF= 000050           F.R4   = 000020          LINC    005102R  025 M.IDN  = 000000          ND.$PN= 001400
F.BKP1= 000051           F.SEQN= 000100          LINE    005016R  025 M.INSP= 000006          NEGSGN  005063R  025
F.BKST= 000024           F.SPDV= 000072          LININC  005026R  025 M.IOTA= 000000          NINPT1  003302R  025
F.BKVB= 000064           F.SPUN= 000074          LISIZE= 000020        M.LIT  = 000205          NINPT2  003300R  025
F.BR   = 000020          F.STBK= 000036          LITOP   005072R  025 M.LN   = 000001          NINSTK  003304R  025
F.CHR  = 000075          F.STK  = 000022          LNLNG  = 104541        M.MMD  = 000010          NINTMP  003310R  025
F.CNTG= 000034           F.STOP= 000004          LOCLCT  005036R  025 M.NEG  = 000000          NINTYP  003306R  025
F.CODE= 000010           F.STP  = 000001          LOWCOR  000002R  025 M.NOT  = 000000          NOTAVL= 104472
F.DFNB= 000046           F.SUSP= 000002          LSTERP  004766R  025 M.NUB  = 000006          NOTOPN= 104473
F.DSPT= 000044           F.TRC  = 000002          LWIDTH  000070R  025 M.NUM  = 000204          NPRT1   004762R  025
F.DVNM= 000134           F.UNIT= 000136          L.BITS= 000003        M.OUTR= 000012          NULL    007364R  025
F.EFBK= 000010           F.URBD= 000020          L.CNT  = 000002        M.PITM= 000000          NUMCHN= 000020
F.EFN  = 000050          F.VAL  = 177776          L.PRNT= 000001        M.PRDS= 000006          NUMDFL  003230R  025
F.END  = 002000          F.VBN  = 000064          L.PTR  = 000000        M.QQAD= 000000          NUMEFL  003233R  025
F.EOBB= 000032           F.VBSZ= 000060          L.PTR1= 000004        M.QUAD= 000000          NUMESN  003236R  025
F.ERR  = 000052          GETCOR= ****** GX        L.SUBS= 000002        M.RAND= 000012          NUMFLG  003230R  025
F.FACC= 000043           GNOME   003102R  025 MACHIN= 000106        M.RAVL= 000000          NUMLEN  000062R  025
F.FFBY= 000014           HASHC   005136R  025 MAXCHN= 000014        M.RCIP= 000000          NUMNEG  003232R  025
F.FLAG= 000002           HASHTB  005140R  025 MONBLK  003120R  025 M.RDCT= 000011          NUMQFL  003234R  025
F.FLB  = 000100          HSHMSK= 177601          MONCVT  003132R  025 M.REVS= 000003          NUMRIG  003231R  025
F.FNAM= 000110           HSHSIZ= 000200          MONLEN  003130R  025 M.ROLL= 000000          NUMWID  003226R  025
```

FNEND MACRO M1113 08-FEB-80 12:24 PAGE 46-3
SYMBOL TABLE

```
NUMZFL  003235R     025 RARGBL  003204R     025 R.OFF  = 000000          SY.CLR= 000040          T.LIT  = 000020
NYIERR= 104432          RARGCV  003220R     025 R.PREV= 000004          SY.ERP= 000020          T.MON  = 000010
N.BLSZ= 000036          RARGFL  003202R     025 R.PTR  = 000000          SY.NOU= 000100          T.NIL  = 000004
N.DESC= 000020          RARGLE  003216R     025 R.SEQ  = 000003          SY.OPN= 000002          T.NUM  = 002000
N.DEV = 000032          RARGLN  003210R     025 R.VAR  = 000002          SY.PMT= 001000          T.OPFN= 000001
N.DEVN= 000034          RARGPT  003206R     025 R6     =%000006          SY.SRT= 000400          T.OTHR= 040000
N.DID = 000024          RARGSF  003214R     025 R7     =%000007          SY.SVC= 000004          T.SOP  = 000002
N.DVNM= 000032          RARGTY  003212R     025 SI     =%000005          SY.SVL= 000200          T.SPOP= 000040
N.EXT = 000014          RARGVA  003222R     025 SIDAMN= 104433           S.AB   = 177776          T.TMP  = 000200
N.FID = 000000          RBASE   005074R     025 SIFRAM  003314R     025 S.ASB  = 100000          T.VAL  = 100000
N.FNAM= 000006          RCOUNR  005112R     025 SISTAK  006350R     025 S.BFHD= 000020          T.VAR  = 001000
N.FTYP= 000014          RCOUNT  005110R     025 SITOP   006346R     025 S.BITS= 000002          UPPER  = 177400
N.FVER= 000016          REDCNT  005070R     025 SITP    007144R     025 S.CHR1= 000003          UTNAME  000344R     023
N.NAME= 000006          REDIGN  005066R     025 SIWARN= 000200           S.FATT= 000016          VALERR= 104413
N.NEXT= 000022          REDOFF  005072R     025 SIZTAB  000044R     025 S.FDB  = 000140          VARBLE  007400R     025
N.PPN  = 000024         REENTR  003060R     025 SI.BIT= 177774           S.FLB  = 040000          VARBLS  000000R     025
N.STAT= 000020          RESBIT  003142R     025 SI.LEN= 177772           S.FNAM= 000006          VERNUM  000000R     025
N.UNIT= 000034          RESCVT  003152R     025 SI.RET= 177772           S.FNB  = 000036          VRSION= 000002
OBUFLN= 001000          RESERR= 104440           SI.VAL= 177776          S.FNBW= 000017          WIDTH   000072R     025
OCHAN   005052R     025 RESLEN  003150R     025 SKPSIZ  005134R     025 S.FNTY= 000004          WROUND  000020R     025
OMODE   000402R     023 RESPTR  003140R     025 SPHOLD  000004R     024 S.FTYP= 000002          WSDATE  000136R     025
OPTMP0  005066R     025 RESSFT  003146R     025 SPINIT  000002R     024 S.LEN  = 000002          WSFULL= 104403
OPTMP1  005070R     025 RESTYP  003144R     025 STKINC= 000774           S.LINK= 177774          WSNAME  000074R     025
OPTMP2  005072R     025 RESULT  003136R     025 STOP   = 104420          S.NFEN= 000020          WSTIME  000132R     025
OPTMP3  005074R     025 RESVAL  003154R     025 SUBCNT  003104R     025 S.TYPE= 140000          $FIS   = ****** G
OPTMP4  005076R     025 RETCOR= ****** GX        SUBDCT  003106R     025 S.VAL  = 000000          $FPP   = ****** G
OPTMP5  005100R     025 RINC    005104R     025 SUBPTR  003110R     025 TKDIMS  005072R     025 $FPPLN= ****** G
OPTMP6  005102R     025 RLTBAS  005066R     025 SYMB1   005344R     025 TKFRAM  005066R     025 $MACHN= ****** G
OPTMP7  005104R     025 ROTINC  005070R     025 SYMF    005340R     025 TKLEFT  005070R     025 $RSTS  = ****** G
OPTMP8  005106R     025 ROUND   000010R     025 SYMPT1  005346R     025 TMODE   000403R     023 $RSX   = ****** G
OPTMP9  005110R     025 ROVER   003312R     025 SYMTAB  005352R     025 TMPTAG= 000166           $RT11  = ****** G
OPTM10  005112R     025 RSTS   = 000000          SYMTBF= 104404          TTYBUF  004140R     025 .BOOLE= 000002
OPTM14  005026R     025 RSX    = 000115          SYMTOP  005350R     025 TTYLEN= 000622           .CHARA= 000000
ORIGIN  000030R     025 RTOP    005070R     025 SYMTP   006272R     025 TURKEY= 104436           .DEF.  = 000002
OUTPTR  000066R     025 RT11   = 000000          SYMWRN= 000054           TYPE   = 000166          .FLOAT= 000006
OVROPR= 000000               R.BITS= 000003          SYNERR= 104407          T.BOL  = 020000          .FSRCB= ****** G
PATCH   007344R     025 R.FIX  = 000001          SYSERR= 104402          T.CON  = 004000          .INTEG= 000004
PRECSN  000052R     025 R.LCT  = 000002          SYSTAT  000000R     024 T.DIM  = 000100          .MOLUN  000404R     023
RANDAC  003064R     025 R.LN   = 000002          SY.BAD= 000001          T.FUN  = 010000          ...GBL= 000000
RANDY   003062R     025 R.LNUM= 000010           SY.BRK= 100000          T.LIST= 000400           ...TPC= 000020
RANERR= 104411               R.NEXT= 000006          SY.BS  = 000010

. ABS.  000000      000
        000000      001
TABLES  000000      002
MONTB   000000      003
DYDTB   000000      004
BOPTB   000000      005
MONBTB  000000      006
DYDBTB  000000      007
COMTB   000000      010
CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
```

```
FNEND   MACRO M1113  08-FEB-80 12:24  PAGE 46-4
SYMBOL TABLE

MNTRTL  000000      017
CODE    000200      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  12929 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:55
FNEND,DM1:[305,301]FNEND/-SP=APLRX1,APLDEF,FNEND,END
```

```
**FILE**ID**FNEXEC
```

```
FFFFFFFFFF   NN      NN   EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC
FFFFFFFFFF   NN      NN   EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC
FF           NN      NN   EE           XX      XX   EE            CC
FF           NN      NN   EE           XX      XX   EE            CC
FF           NNNN    NN   EE             XX  XX     EE            CC
FF           NNNN    NN   EE             XX  XX     EE            CC
FFFFFFFF     NN  NN  NN   EEEEEEEE         XX       EEEEEEEE      CC
FFFFFFFF     NN  NN  NN   EEEEEEEE         XX       EEEEEEEE      CC
FF           NN    NNNN   EE             XX  XX     EE            CC
FF           NN    NNNN   EE             XX  XX     EE            CC
FF           NN      NN   EE           XX      XX   EE            CC                ....
FF           NN      NN   EE           XX      XX   EE            CC                ....
FF           NN      NN   EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC         ....
FF           NN      NN   EEEEEEEEEE   XX      XX   EEEEEEEEEE     CCCCCCCC         ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

.MAIN. MACRO M1113 08-FEB-80 12:27
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGF
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    FNCALL           CALL USER DEFINED FUNCTION
```

FNEXEC MACRO M1113 08-FEB-80 12:27
TABLE OF CONTENTS

47- 173   FNEXEC   EXECUTE USER-DEFINED FUNCTION
49- 373   FNEXIT   EXIT FROM FUNCTION EXECUTION

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 1


      1          000000                RT11    =  0          ;NOT RT-11
      2          000000                RSTS    =  0          ;NOT RSTS/E
      3          000115                RSX     = 'M          ;RSX-11/M
      4
      5          000000                EIS     =  0          ;NO EIS
      6          000000                FIS     =  0          ;NO FIS
      7          000000                FPP     =  0          ;NO FPP
      8
      9          000004                FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11          000000                DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 12:27  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

DEFINITIONS   ERRORS

```
                                .TITLE  FNEXEC
                                .IDENT  \V02.03\

                                .SBTTL  FNCALL          CALL USER DEFINED FUNCTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'MONAD', 'DYAD', and 'VAL' call 'FNCALL' when the evaluation
                                ;       of a user-defined function is desired.  When a function is
                                ;       executed, the values of the local variables must be saved
                                ;       (on the SI stack in the F-frame) so that when the function
                                ;       is exited ('FNEXIT') the old values of the local variables
                                ;       (and labels) can be restored.  The argument(s) of a function
                                ;       and the result are also local variables.
                                ;         This routine creates an F-frame on the SI stack and places
                                ;       in it the old values (actually the value poiners) of the local
                                ;       variables.  When this is done, the arguments are bound to the
                                ;       dummy variables which are the arguments of the function, and
                                ;       'FNEXEC' is branched to and the function is executed.
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    FNCALL
                                ;
                                ; INPUTS:
                                ;       R4      - Points to name block of function to be evaluated.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       An F-frame is created on the SI stack for the function call,
                                ;       and local variables are handled.
                                ;
                                ; ROUTINES CALLED:
                                ;       ASSIGN  - Assign an arg to a variable
                                ;       BRKCHK  - Check for CONTROL-C interrupt
                                ;       FNEXIT  - End the function, nothing to do
                                ;       STKXTN  - Not enough room left on SI, get more
                                ;--

000000                          .PSECT  CODE
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 45
FNCALL          CALL USER DEFINED FUNCTION

     43 000000                              FNCALL::
     44 000000                                      CALL    BRKCHK          ;CHECK TO SEE IF HE'S HAD ENOUGH OF APL
     45 000004  010502                              MOV     SI,R2           ;SAVE POINTER TO ARGS
     46 000006  011403                              MOV     @R4,R3          ;POINT TO FN HEADER
     47 000010  116300  000002                      MOVB    R.LCT(R3),R0    ;GET LOCAL VARIABLE COUNT
     48 000014  062700  000003                      ADD     #3,R0           ;PLUS ARGS
     49 000020  062703  000010                      ADD     #R.LNUM,R3      ;POINT PAST
     50 000024  063703  000062'                     ADD     NUMLEN,R3       ;LINE NUMBER
     51 000030  011303                              MOV     @R3,R3          ;POINT TO LABEL BLOCK
     52 000032  001403                              BEQ     12$             ;NO LABELS
     53 000034  116303  000002                      MOVB    L.CNT(R3),R3    ;GET COUNT OF LABELS
     54 000040  060300                              ADD     R3,R0           ;TOTAL STACK SIZE NEEDED
     55 000042  006300                      12$:    ASL     R0              ;IN WORDS
     56 000044  060500                              ADD     SI,R0           ;END OF FRAME
     57 000046  023700  006346'                     CMP     SITOP,R0        ;TOO MUCH?
     58 000052  101002                              BHI     10$             ;NOPE
     59 000054                                      CALL    STKXTN          ;EXTEND THE STACK
     60 000060  013700  003314'             10$:    MOV     SIFRAM,R0       ;POINT TO PREVIOUS FRAME
     61 000064  013760  004764'  000002            MOV     FLAGS,F.FLAG(R0);SAVE THE FRAME'S FLAGS
     62 000072  012625                              MOV     (SP)+,(SI)+     ;SAVE RETURN TO CALLING ROUTINE
     63 000074  005025                              CLR     (SI)+           ;CLEAR PARSE FRAME'S BITS
     64 000076  005025                              CLR     (SI)+           ;NO RESULT OF FN YET
     65 000100  010501                              MOV     SI,R1           ;COPY SI POINTER
     66 000102  010025                              MOV     R0,(SI)+        ;POINT TO PREVIOUS FRAME
     67 000104  010137  003314'                     MOV     R1,SIFRAM       ;SAVE POINTER TO THIS F-FRAME
     68 000110  012737  001000  004764'             MOV     #F$FUNC,FLAGS   ;THIS IS AN F-FRAME
     69 000116  005025                              CLR     (SI)+           ;CLEAR FRAME'S FLAGS
     70 000120  005025                              CLR     (SI)+           ;CLEAR LAST C-BYTE
     71 000122  005025                              CLR     (SI)+           ;CLEAR HEADER POINTER
     72 000124  011403                              MOV     @R4,R3          ;GET POINTER TO FN HEADER
     73 000126  001001                              BNE     2$              ;IS IT REALLY A FUNCTION?
     74 000130  104402                      1$:     SYSERR                  ;NOPE, FN CALL WITHOUT FUNCTION?
     75 000132  010325                      2$:     MOV     R3,(SI)+        ;SAVE POINTER TO FN HEADER
     76 000134  010325                              MOV     R3,(SI)+        ;HEADER IS INITIAL LINE
     77 000136  012625                              MOV     (SP)+,(SI)+     ;COPY INPUT CODESTRING PTR (R2)
     78 000140  012625                              MOV     (SP)+,(SI)+     ;SAVE CURRENT TOKEN BITS   (R3)
     79 000142  012625                              MOV     (SP)+,(SI)+     ;SAVE CURRENT TOKEN VALUE  (R4)
     80 000144  012625                              MOV     (SP)+,(SI)+     ;CALLING ROUTINE LINK WORD
     81 000146                                      PUSH    R2              ;SAVE POINTER TO ARGUMENTS
     82 000150  010302                              MOV     R3,R2           ;COPY POINTER TO FN HEADER
     83 000152  062702  000010                      ADD     #R.LNUM,R2      ;POINT TO LINE # OF HEADER (0)
     84 000156  063702  000062'                     ADD     NUMLEN,R2       ;MOVE PAST LINE # TO LABEL POINTER
     85 000162                                      PUSH    R2              ;SAVE PTR INTO HEADER
     86 000164  005722                              TST     (R2)+           ;SKIP PAST LABEL POINTER
     87 000166  012701  000003                      MOV     #3,R1           ;Get number of result, left and right args
     88 000172  012200                      FNCAL1: MOV     (R2)+,R0        ;Argument's name block
     89 000174  001403                              BEQ     1$              ;There isn't one
     90 000176  011025                              MOV     (R0),(SI)+      ;Save its old value in the F-frame
     91 000200  005010                              CLR     (R0)            ;It has no new value yet
     92 000202  000401                              BR      2$              ;Go to loop
     93 000204  005025                      1$:     CLR     (SI)+           ;Clear argument pointer in F-frame
     94 000206                              2$:     SOB     R1,FNCAL1       ;Loop for all arguments
     95 000212  116301  000002                      MOVB    R.LCT(R3),R1    ;GET LOCAL VARIABLE COUNT
     96 000216  001406                              BEQ     10$             ;There aren't any
     97 000220  012200                      3$:     MOV     (R2)+,R0        ;POINT TO LOCAL VAR'S NAME BLOCK
     98 000222  001402                              BEQ     4$              ;NO LOCAL, SO SKIP
     99 000224  011025                              MOV     @R0,(SI)+       ;PUSH ITS OLD VALUE
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 45-1
FNCALL           CALL USER DEFINED FUNCTION

    100 000226  005010                          CLR     @R0             ;NOW LOCAL HAS NO VALUE
    101 000230                          4$:     SOB     R1,3$           ;SAVE ALL LOCAL VARIABLES
```

```
                                        ;
                                        ;THE LOCAL VARIABLES HAVE BEEN INITIALIZED, SO
                                        ;SET UP THE LABELS FOR THIS FUNCTION.
                                        ;
000234  017602  000000          10$:    MOV     @0(SP),R2       ;POINT TO LABEL LIST BLOCK
000240  001420                          BEQ     6$              ;GOOD FOR HIM...NO LABLES
000242  116201  000002                  MOVB    L.CNT(R2),R1    ;GET LABEL COUNT
000246  001001                          BNE     14$             ;There better be some
000250  104402                          SYSERR                  ;No labels, Then there shouldn't be a label block
000252  012704  000010          14$:    MOV     #R.LNUM,R4      ;GET OFFSET TO LINE #
000256  063704  000062'                  ADD     NUMLEN,R4       ;GET OFFSET TO LABEL PTR IN LINE BLOCK
000262  011202                          MOV     @R2,R2          ;POINT TO FIRST POINER

000264  012200                  5$:     MOV     (R2)+,R0        ;GET PTR TO LINE BLOCK
000266  060400                          ADD     R4,R0           ;POINT TO LABEL POINTER
000270  012003                          MOV     (R0)+,R3        ;GET LABEL'S NAME BLOCK POINTER
000272  011325                          MOV     @R3,(SI)+       ;SAVE LABEL'S OLD VALUE
000274  010013                          MOV     R0,@R3          ;LABEL'S VALUE IS LINE #
000276                                  SOB     R1,5$           ;PROCESS ALL LABELS
                                        ;
                                        ;NOW THAT LABELS ARE INITIALIZED, SET UP THE LEFT
                                        ;AND RIGHT ARGUMENTS FOR THE FUNCTION...I.E. ASSIGN
                                        ;VALUES TO THEM.
                                        ;
000302                          6$:     POP     R2              ;RESTORE PTR INTO FN HEADER
000304  062702  000004                  ADD     #4,R2           ;POINT TO LEFT ARG
000310                                  POP     R1              ;POINT TO LEFT ARG
000312  012204                          MOV     (R2)+,R4        ;GET LEFT ARG'S NAME
000314  001427                          BEQ     7$              ;IF NONE THEN SKIP
000316                                  PUSH    SI.BITS(R1)     ;SAVE SI BITS
000322  016103  177776                  MOV     SI.VAL(R1),R3   ;GET 'OLD' POINTER TO DATA
000326  132763  000010  000003          BITB    #A.TMP,A.BITS(R3) ;IS IT A TEMPORARY?
000334  001401                          BEQ     12$             ;YES, DON'T WORRY
000336  005016                          CLR     (SP)            ;NO, SO KILL THE BITS
000340                          12$:    PUSH    R1              ;SAVE SI POINTER
000342                                  CALL    ASSIGN          ;ASSIGN LEFT ARG TO VARIABLE
000346                                  POP     <R1,SI.BITS(R1)>;GET BACK PTR TO VAL FRAME
000354  020061  177776                  CMP     R0,SI.VAL(R1)   ;WAS ARG COPIED?
000360  001003                          BNE     11$             ;YES, SO WE MUST DEALLOCATE COPY
000362  152760  000040  000003          BISB    #A.ARG,A.BITS(R0);NO, PREVENT DEALLOCATION
000370  062701  177764          11$:    ADD     #2*SI.LEN,R1    ;POINT TO RIGHT ARG'S VAL FRAME
000374  012204                  7$:     MOV     (R2)+,R4        ;POINT TO RIGHT ARG'S VALUE
000376  001425                          BEQ     8$              ;IF NONE THEN SKIP
000400                                  PUSH    SI.BITS(R1)     ;SAVE BITS
000404  016103  177776                  MOV     SI.VAL(R1),R3   ;POINT TO VALUE
000410  132763  000010  000003          BITB    #A.TMP,A.BITS(R3) ;IS IT ASSIGNED?
000416  001401                          BEQ     13$             ;NOPE
000420  005016                          CLR     (SP)            ;YES, CLEAR SI BITS
000422                          13$:    PUSH    R1              ;SAVE POINTER TO ARG'S FRAME
000424                                  CALL    ASSIGN          ;ASSIGN VALUE TO RIGHT ARG
000430                                  POP     <R1,SI.BITS(R1)>;RESTORE POINTER TO FRAME
000436  020061  177776                  CMP     R0,SI.VAL(R1)   ;WAS ARG COPIED?
000442  001003                          BNE     8$              ;YES, SO WE MUST DEALLOCATE COPY
000444  152760  000040  000003          BISB    #A.ARG,A.BITS(R0);ELSE MARK BLOCK TO PREVENT DEALLOCATION BY US
                                        ;
                                        ;INITIALIZATION IS COMPLETE.  POINT TO FIRST LINE IN
                                        ;FUNCTION AND START FUNCTION EXECUTION.
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 46-1
FNCALL           CALL USER DEFINED FUNCTION

    160
    161 000452  013703  003314'          8$:     MOV     SIFRAM,R3           ;POINT TO THIS F-FRAME
    162 000456  010563  000004                   MOV     SI,F.LSTC(R3)       ;POINT TO END OF SI
    163 000462  016302  000010                   MOV     F.FUN(R3),R2        ;POINT TO FN HEAD
    164 000466  016204  000006                   MOV     R.NEXT(R2),R4       ;POINT TO FIRST LINE OF FN
    165 000472  020402                           CMP     R4,R2               ;NO LINES IN FN?
    166 000474  001402                           BEQ     9$                  ;NOPE!?!
    167 000476  000137  000512'                  JMP     FNEXEC              ;START FUNCTION EXECUTION
    168
    169 000502                           9$:     CALL    FNEXIT              ;ELSE POP F-FRAME
    170 000506  000175  177772                   JMP     @SI.RET(SI)         ;AND RETURN TO CALLER
    171
```

```
                                .SBTTL  FNEXEC          EXECUTE USER-DEFINED FUNCTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'FNEXEC' controls the execution of user-defined functions.
                                ;       it runs through the function executing one line at a time
                                ;       until a stop or trace is encountered on a line.  This routine
                                ;       also handles branches to different lines in the function.
                                ;       A bad branch or hitting the end of the function causes
                                ;       'FNEXIT' to be called, and a return is made to the original
                                ;       caller with the result of the function (if any) in a parse
                                ;       frame at the end of the SI stack.
                                ;
                                ; CALLING SEQUENCE:
                                ;       JMP     FNEXEC
                                ;
                                ; INPUTS:
                                ;       R4      - Points to line block of next line to execute.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       EXPR    - Evaluate expression in line
                                ;       FNIDEN  - Identify function (Trace/Stop)
                                ;       GETNXT  - Get next token
                                ;       OUTCHR  - Print a character
                                ;       PRINT   - Print the result
                                ;--

000512                          .PSECT  CODE


000512                          FNEXEC::
000512  013703  003314'                MOV     SIFRAM,R3       ;POINT TO THIS F-FRAME
000516  052737  100000  004764'        BIS     #F.PRNT,FLAGS   ;PRINT LINE'S RESULT
000524  010463  000012              MOV     R4,F.LINE(R3)   ;SAVE POINTER TO CURRENT LINE
000530  020437  005046'              CMP     R4,BRSELF       ;BRANCHING TO SELF?
000534  001417                      BEQ     10$             ;YES, DON'T STOP
000536  132764  000001  000003        BITB    #F.STP,R.BITS(R4);STOP VECTOR SET?
000544  001413                      BEQ     10$             ;NO, FINE
000546                              CALL    NEWLIN          ;Skip to new line
000552                              CALL    FNIDEN          ;IDENTIFY LINE
000556  052737  000002  004764'        BIS     #F.SUSP,FLAGS   ;SET SUSPENDED BIT
000564                              CALL    NEWLIN          ;Skip to New Line
000570  000137  000000G              JMP     ESPAWN          ;AND SPAWN AN E FRAME

000574  005037  005046'        10$:    CLR     BRSELF          ;NO LONGER ALLOW BRANCHES TO SELF
000600  010402                      MOV     R4,R2           ;COPY PTR TO LINE BLOCK
000602  061202                      ADD     @R2,R2          ;POINT PAST CODESTRING END
                                .ENABL LSB
000604                          FNEXE0: CALL    GETNXT          ;GET FIRST TOKEN
000610                              CALL    EXPR            ;EVALUATE THE EXPRESSION
000614  042737  000020  004764'        BIC     #F.BR,FLAGS     ;CLEAR BRANCH FLAG
000622  032703  020000              BIT     #T.BOL,R3       ;END OF EXPRESSION?
000626  001031                      BNE     FNEXE1          ;YES, NORMAL TERMINATION
000630  032703  040000              BIT     #T.OTHR,R3      ;NO, DID IT END WITH A CHAR?
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 47-1
FNEXEC          EXECUTE USER-DEFINED FUNCTION

    230 000634  001001                          BNE     2$              ;YES, CHECK FOR A BRANCH
    231 000636  104407                  1$:     SYNERR                  ;ILLEGAL EXPRESSION TERMINATION
    232 000640  122704  000135          2$:     CMPB    #CHRFRO,R4      ;BRANCH?
    233 000644  001407                          BEQ     21$             ;YES, GOOD
    234 000646  122704  000074                  CMPB    #CHRSEM,R4      ;SEMI-COLON?
    235 000652  001371                          BNE     1$              ;NO, BAD EXPRESSION
    236 000654  052737  100000  004764'        BIS     #F.PRNT,FLAGS   ;PRINT LINE'S RESULT
    237 000662  000750                          BR      FNEXE0          ;ELSE SPECIAL OUTPUT FORMATTING
    238 000664                          21$:    CALL    GETNXT          ;BRANCH LAST THING ON LINE?
    239 000670  032703  020000                  BIT     #T.BOL,R3       ;END OF LINE?
    240 000674  001760                          BEQ     1$              ;NO, BRANCH INSIDE EXPRESSION ILLEGAL
    241 000676  052737  000020  004764'        BIS     #F.BR,FLAGS     ;SET BRANCH FLAG
    242 000704  042737  100000  004764'        BIC     #F.PRNT,FLAGS   ;AND CLEAR PRINT FLAG
    243                                 ;
    244                                 ;CHECK FOR TRACE OF FUNCTION LINE
    245                                 ;
    246 000712  013703  003314'          FNEXE1: MOV     SIFRAM,R3       ;POINT TO THIS F-FRAME
    247 000716  016304  000012                  MOV     F.LINE(R3),R4   ;POINT TO CURRENT LINE BLOCK
    248 000722  132764  000002  000003          BITB    #F.TRC,R.BITS(R4);TRACE THE LINE OR STOP?
    249 000730  001406                          BEQ     4$              ;NO, DON'T IDENTIFY THE FUNCTION
    250 000732                                  CALL    FNIDEN          ;IDENTIFY FUNCTION AND LINE NUMBER
    251 000736  112700  000040                  MOVB    #CHRSPA,R0      ;GET A SPACE
    252 000742                                  CALL    OUTCHR          ;PRINT IT
    253 000746  116400  000003          4$:     MOVB    R.BITS(R4),R0   ;GET LINE'S BITS
    254 000752  053700  004764'                 BIS     FLAGS,R0        ;ADD IN FLAG BITS
    255 000756  032700  100002                  BIT     #F.TRC+F.PRNT,R0;PRINT LINE'S RESULT?
    256 000762  001434                          BEQ     5$              ;NO
    257 000764  010501                          MOV     SI,R1           ;COPY SI POINTER
    258 000766  016302  000004                  MOV     F.LSTC(R3),R2   ;POINT BEFORE FIRST PARSE FRAME
    259 000772  013700  005052'                 MOV     OCHAN,R0        ;POINT TO OUTPUT CHANNEL
    260 000776  016046  177764                  MOV     C.CPTR(R0),-(SP);SAVE OUTPUT POINTER
    261
    262 001002  020102                  13$:    CMP     R1,R2           ;DONE YET?
    263 001004  003002                          BGT     14$             ;NO, PRINT IT
    264 001006  001407                          BEQ     15$             ;YES, DONE PRINTING
    265 001010  104402                          SYSERR                  ;STACK SCREWED!
    266
    267 001012  010100                  14$:    MOV     R1,R0           ;COPY FRAME POINTER
    268 001014                                  CALL    PRINT           ;PRINT RESULT OF LINE
    269 001020  062701  177772                  ADD     #SI.LEN,R1      ;BACK UP 1 FRAME
    270 001024  000766                          BR      13$             ;AND CHECK IF MORE TO PRINT
    271
    272 001026  132764  000002  000003  15$:    BITB    #F.TRC,R.BITS(R4);TRACING FUNCTION EXECUTION?
    273 001034  001005                          BNE     16$             ;YES, SO ALWAYS PRINT CRLF
    274 001036  013700  005052'                 MOV     OCHAN,R0        ;POINT TO OUTPUT CHANNEL
    275 001042  026026  177764                  CMP     C.CPTR(R0),(SP)+;DID WE PRINT ANYTHING?
    276 001046  001402                          BEQ     5$              ;NO, SO DON'T PRINT CR/LF!
    277 001050                          16$:    CALL    NEWLIN          ;Skip to new line
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 48
FNEXEC          EXECUTE USER-DEFINED FUNCTION


                                        ;
                                        ;CHECK FOR BRANCH, AND IF NOT GET AND EXECUTE NEXT LINE.
                                        ;
001054  032737  000020  004764' 5$:     BIT     #F.BR,FLAGS     ;BRANCH FLAG SET?
001062  001015                         BNE     FNBR            ;YES, SPECIAL HANDLING

001064                          FNEXE6: CALL    PKILL           ;NO, KILL RESULT OF LINE
001070  016404  000006                  MOV     R.NEXT(R4),R4   ;POINT TO NEXT LINE
001074  020463  000010                  CMP     R4,F.FUN(R3)    ;BACK TO FN HEADER?
001100  001402                         BEQ     FNDONE          ;YES EXIT THE FUNCTION
001102  000137  000512'                JMP     FNEXEC          ;ELSE EXECUTE THE NEXT LINE

001106                          FNDONE::
001106                                  CALL    FNEXIT          ;YES, KILL THE F-FRAME
001112  000175  177772                  JMP     @SI.RET(SI)     ;AND RETURN TO CALLER
                                        ;
                                        ;PROCESS A BRANCH REQUEST.  NULL BRANCH IGNORED.
                                        ;A ZERO BRANCH OR BRANCH PAST FN END CAUSES EXIT
                                        ;FROM FN.
                                        ;
001116                          FNBR::
001116  032765  100000  177774          BIT     #T.VAL,SI.BITS(SI);EXPRESSION HAVE A VALUE?
001124  001012                         BNE     FNBR2           ;YES
001126  032737  001100  004764'        BIT     #F$EVAL+F$FUNC,FLAGS;FUNCTION OR USER LEVEL?
001134  001004                         BNE     22$             ;YES, CLEAR SI SOMEWHAT
001136                                  PUSH    SI              ;PUSH A NON-ZERO VALUE ON THE STACK
001140  005037  003134'                CLR     MONVAL          ;EQUIV TO BRANCH TO LINE 0.
001144  000420                         BR      FNBR3           ;AND GO
001146  000137  000000G         22$:    JMP     CLRSI           ;NO, KILL ALL PENDANT FUNCTIONS!!

001152  016502  177776          FNBR2:  MOV     SI.VAL(SI),R2   ;POINT TO SOURCE
001156  116200  000003                  MOVB    A.BITS(R2),R0   ;GET BITS
001162  002141                         BGE     FNBSYS          ;ERROR
001164  042700  177771                  BIC     #^C<A.NUM>,R0   ;CLEAR GARBAGE BITS
001170  001537                         BEQ     FNBDOM          ;DOMAIN ERROR, IS CHARACTER
001172                                  PUSH    A.ALEN(R2)      ;IS ARG A NULL??
001176  001403                         BEQ     FNBR3           ;YES
001200  011202                         MOV     (R2),R2         ;POINT TO DATA
001202                                  CALL    @IMONCT(R0)     ;FETCH THE BRANCH VALUE
001206                          FNBR3:  CALL    PKILL           ;KILL RESULT OF LINE
001212  032737  000300  004764'        BIT     #F$EVAL+F$EVI,FLAGS;IMMEDIATE MODE?
001220  001443                         BEQ     25$             ;NO, OTHERWISE DELETE TOP FRAME
001222  032737  000100  004764'        BIT     #F$EVAL,FLAGS   ;NEED TO RESTORE POINTERS?
001230  001012                         BNE     23$             ;NO
001232  013703  003314'                MOV     SIFRAM,R3       ;INPUT PTRS MUST BE RESET
001236  016303  000006                  MOV     F.HEAD(R3),R3   ;POINT PAST HEADED DATA
001242  014337  005056'                MOV     -(R3),E$CT      ;RESTORE COUNT
001246  014337  005060'                MOV     -(R3),E$PT      ;         POINTER
001252  014337  000064'                MOV     -(R3),INPTR     ;         INPUT POINTER
001256  013705  003314'          23$:  MOV     SIFRAM,SI       ;POINT TO OLD REF POINT
001262  011503                         MOV     (SI),R3         ;POINT TO PREVIOUS FRAME
001264  010337  003314'                MOV     R3,SIFRAM       ;SAVE PREVIOUS FRAME POINTER
001270  001005                         BNE     24$             ;REAL POINTER?
001272  005726                         TST     (SP)+           ;ADJUST STACK
001274  062705  177772                  ADD     #SI.LEN,SI      ;ADJUST FOR THE TURKEYS
001300  000137  000000G                 JMP     ESPAWN          ;AT THE BOTTOM, FINISHED
001304  016337  000002  004764' 24$:    MOV     F.FLAG(R3),FLAGS;RESTORE FLAGS
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 48-1
FNEXEC          EXECUTE USER-DEFINED FUNCTION

    336 001312  032737  000300  004764'          BIT     #F$EVAL+F$EVI,FLAGS;IMMEDIATE MODE?
    337 001320  001403                           BEQ     25$             ;NO
    338 001322  005726                           TST     (SP)+           ;ADJUST THE STACK
    339 001324  000137  000000G                  JMP     EXEC            ;AND CONTINUE
    340 001330  032737  000400  004764' 25$:     BIT     #F$EXEC,FLAGS   ;EPSILON ?
    341 001336  001431                           BEQ     26$             ;NO
    342 001340  032737  002000  004764'          BIT     #F.END,FLAGS    ;Branch at beginning of line?
    343 001346  001451                           BEQ     FNBSYN          ;No, then SYNTAX error
    344 001350  010516                           MOV     SI,(SP)         ;SAVE FOR LATER
    345 001352  013705  003314'                  MOV     SIFRAM,SI       ;POINT TO OLD REF PT
    346 001356  011503                           MOV     (SI),R3         ;POINT TO DATA
    347 001360  016505  000006                   MOV     F.HEAD(SI),SI   ;POINT PAST HEADER
    348 001364  014537  005056'                  MOV     -(SI),E$CT      ;RESTORE COUNT
    349 001370  014537  005060'                  MOV     -(SI),E$PT      ;         POINTER
    350 001374  014537  000064'                  MOV     -(SI),INPTR     ;         INPUT POINTER
    351 001400                                   POP     SI              ;RESTORE SI
    352 001402                                   SIFRAM  #NULL,#T.NUM+T.CON;Numeric constant
    353 001416  000137  000000G                  JMP     EVAL1           ;AND GO
    354 001422  005726                  26$:     TST     (SP)+           ;AND THE LEN WAS...
    355 001424  001617                           BEQ     FNEXE6          ; 0 LONG
    356 001426  005737  003134'                  TST     MONVAL          ;CHECK LINE NUMBER
    357 001432  003625                           BLE     FNDONE          ;<=0 MEANS EXIT FUNCTION
    358 001434  016302  000010                   MOV     F.FUN(R3),R2    ;POINT TO FUNCTION HEADER BLOCK
    359 001440  010204                           MOV     R2,R4           ;COPY POINTER
    360 001442  016404  000006          FNBR1:   MOV     R.NEXT(R4),R4   ;POINT TO NEXT LINE BLOCK
    361 001446  020402                           CMP     R4,R2           ;BACK TO FUNCTION HEAD?
    362 001450  001616                           BEQ     FNDONE          ;IF SO THEN DONE WITH FUNCTION
    363 001452  026437  000010  003134'          CMP     R.LNUM(R4),MONVAL ;LINE NUMBERS MATCH?
    364 001460  001370                           BNE     FNBR1           ;NO, TRY NEXT LINE
    365 001462  000137  000512'                  JMP     FNEXEC          ;YES, START EXECUTING NEW LINE
    366
    367 001466  104402                  FNBSYS:  SYSERR                  ;SYSTEM ERROR
    368 001470  104417                  FNBDOM:  DOMERR                  ;DOMAIN ERROR
    369 001472  104407                  FNBSYN:  SYNERR                  ;Syntax Error
    370
    371                                          .DSABL LSB
```

```
                                .SBTTL  FNEXIT          EXIT FROM FUNCTION EXECUTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       'FNEXIT' is the inverse of 'FNCALL'.  It restores all local
                                ;       variables to their original values, pops off the F-frame on
                                ;       the SI stack, and returns with a parse frame on top of the
                                ;       SI stack containing the result (if any) of the function.
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    FNEXIT
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; ROUTINES CALLED:
                                ;       KILL    - Delete arg on stack
                                ;       RETCOR  - Free space used for locals and whatnot
                                ;--
                                ;
                                ;REGISTER USAGE:
                                ;       R0      - VALUE BLOCK POINTER
                                ;       R1      - COUNT
                                ;       R2      - SYMBOL NAME BLOCK POINTER
                                ;       R3      - HEADER/LABEL LIST POINTER
                                ;       R4      - MAGIC LABEL OFFSET

001474                          .PSECT  CODE
```

```
                                ;
001474                  FNEXIT::
001474  013704  003314'          MOV     SIFRAM,R4       ;POINT TO THE F-FRAME
001500  016403  000004            MOV     F.LSTC(R4),R3   ;POINT PAST END OF F-FRAME
001504  001407                    BEQ     3$              ;NOTHING TO KILL
                                ;
001506  020503          1$:       CMP     SI,R3           ;ANY PARSE FRAMES LEFT?
001510  001405                    BEQ     3$              ;NO
001512  101001                    BHI     2$              ;AT END OF F-FRAME?
001514  104402                    SYSERR                  ;NO, SI STACK SCREWED UP
001516                  2$:       CALL    KILL            ;WIPE OUT THE PARSE FRAME
001522  000771                    BR      1$              ;AND KILL ALL OF THEM LEFT
                                ;+
                                ; MOVE THE RESULT (IF ANY) INTO THE PARSE FRAME AT THE FRONT
                                ; OF THE F-FRAME.
                                ;-
001524  016403  000010  3$:       MOV     F.FUN(R4),R3    ;POINT TO FUNCTION HEADER
001530  010401                    MOV     R4,R1           ;COPY POINTER TO PARSE FRAME
001532  012704  000010            MOV     #R.LNUM,R4      ;GET OFFSET TO LINE NUMBER
001536  063704  000062'            ADD     NUMLEN,R4       ;CALCULATE MAGIC LABEL OFFSET
001542  060403                    ADD     R4,R3           ;POINT TO LABEL LIST PTR IN FN HEAD
001544                            PUSH    R3              ;SAVE POINTER
001546  005723                    TST     (R3)+           ;SKIP TO RESULT NAME POINTER
001550  011303                    MOV     @R3,R3          ;POINT TO RESULT'S NAME BLOCK
001552  001423                    BEQ     5$              ;FUNCTION DOESN'T RETURN RESULT
001554  011302                    MOV     @R3,R2          ;POINT TO VALUE OF RESULT
001556  001421                    BEQ     5$              ;NEVER ASSIGNED A VALUE
001560  005762  000002            TST     A.DIMS(R2)      ;IN SPACEWAR MODE?
001564  002154                    BGE     TURK            ;YES, REFRESH SPACE SHIPS
001566  142762  000010  000003    BICB    #A.TMP,A.BITS(R2);RESULT IS A TEMPORARY!
001574  010241                    MOV     R2,-(R1)        ;PUT VALUE POINTER IN PARSE FRAME
001576  012700  102000            MOV     #T.VAL+T.NUM,R0 ;GET RESULT BITS
001602  132762  000006  000003    BITB    #A.NUM,A.BITS(R2);IS RESULT NUMERIC?
001610  001002                    BNE     4$              ;YES, FINE
001612  042700  002000            BIC     #T.NUM,R0       ;KILL THE NUMERIC BIT
001616  010041          4$:       MOV     R0,-(R1)        ;SET BITS IN PARSE FRAME
001620  005013                    CLR     @R3             ;UNDEFINE RESULT VARIABLE
                                ;+
                                ; RESTORE PREVIOUS VALUES OF THE LABELS
                                ;-
001622  017603  000000  5$:       MOV     @0(SP),R3       ;GET POINTER TO LABEL LIST BLOCK
001626  001423                    BEQ     8$              ;NO LABELS!!
001630  116301  000002            MOVB    L.CNT(R3),R1    ;GET COUNT OF LINES WITH LABELS
001634  011303                    MOV     @R3,R3          ;POINT TO FIRST LINE POINTER
001636  060103                    ADD     R1,R3           ;POINT TO END OF POINTERS
001640  060103                    ADD     R1,R3           ;WORDS NOT BYTES!
                                ;
001642  014300          6$:       MOV     -(R3),R0        ;GET PTR TO NEXT LINE WITH LABEL
001644  060400                    ADD     R4,R0           ;POINT TO LABEL POINTER
001646  012002                    MOV     (R0)+,R2        ;GET PTR TO LABEL'S NAME BLOCK
001650  021200                    CMP     @R2,R0          ;DID LABEL CHANGE?
001652  001406                    BEQ     7$              ;NO, DONE WITH THIS LABEL
001654  011200                    MOV     @R2,R0          ;GET LABEL'S NEW VALUE
001656  005760  000002            TST     A.DIMS(R0)      ;CHECK IF VALUE
001662  002115                    BGE     TURK            ;IT'S NOT, PROBLEM
001664                            CALL    RETCOR          ;KILL LABEL IF VALUE RE-ASSIGNED
001670  014512          7$:       MOV     -(SI),@R2       ;RESTORE LABEL'S OLD VALUE
001672                            SOB     R1,6$           ;AND LOOP TO RESTORE ALL LABELS
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 51
FNEXIT          EXIT FROM FUNCTION EXECUTION


                                        ;
                                        ;RESTORE ALL LOCAL VARIABLES AND FUNCTION ARGUMENT(S)
                                        ;
001676  013703  003314'         8$:     MOV     SIFRAM,R3       ;POINT TO THIS F-FRAME
001702  016303  000010                  MOV     F.FUN(R3),R3    ;POINT TO FUNCTION HEADER BLOCK
001706  116301  000002                  MOVB    R.LCT(R3),R1    ;GET COUNT OF LOCAL VARS
001712                                  POP     R3              ;POINT TO LABEL POINTER
001714  062703  000010                  ADD     #10,R3          ;POINT TO FIRST LOCAL VARIABLE
001720  005701                          TST     R1              ;ANY LOCALS?
001722  001413                          BEQ     12$             ;NO, GOOD
001724  060103                          ADD     R1,R3           ;POINT PAST LAST LOCAL
001726  060103                          ADD     R1,R3           ;WORDS, NOT BYTES!!

001730  014300                  9$:     MOV     -(R3),R0        ;GET NEXT LOCAL NAME POINTER
001732  001405                          BEQ     11$             ;NO LOCAL
001734  005710                          TST     @R0             ;Check its current value
001736  001402                          BEQ     10$             ;HAD NO VALUE
001740                                  CALL    ERASE           ;RELEASE OLD VALUE
001744  014510                  10$:    MOV     -(SI),(R0)      ;RESTORE CURRENT VALUE
001746                          11$:    SOB     R1,9$           ;AND PROCESS NEXT LOCAL

001752  012701  000002          12$:    MOV     #2,R1           ;COUNT FOR LEFT AND RIGHT ARGS
001756  005745                  14$:    TST     -(SI)           ;Point to next argument
001760  014302                          MOV     -(R3),R2        ;POINT TO LEFT/RIGHT ARG
001762  001422                          BEQ     17$             ;NO ARG
001764  011200                          MOV     @R2,R0          ;GET CURRENT VALUE
001766  001417                          BEQ     16$             ;HAS NO VALUE?
001770  020015                          CMP     R0,(SI)         ;SAME AS OLD VALUE?
001772  001412                          BEQ     15$             ;YES
001774  005760  000002                  TST     A.DIMS(R0)      ;SPACEWAR MODE?
002000  002046                          BGE     TURK            ;YES, EVADE THE SUN!
002002  132760  000040  000003          BITB    #A.ARG,A.BITS(R0);DEALLOCATION PROOF ARG?
002010  001003                          BNE     15$             ;YES, SOMEONE ELSE WILL KILL IT
002012                                  CALL    RETCOR          ;NO, RELEASE CURRENT VALUE
002016  000403                          BR      16$             ;AND RESTORE OLD VALUE
002020  142760  000050  000003  15$:    BICB    #A.TMP+A.ARG,A.BITS(R0);CURRENT VALUE IS NOW TEMP
002026  011512                  16$:    MOV     (SI),(R2)       ;RESTORE OLD VALUE
002030                          17$:    SOB     R1,14$          ;AND PROCESS NEXT ARG
002034  005745                          TST     -(SI)           ;Point to Result
002036  014302                          MOV     -(R3),R2        ;POINT TO RESULT'S NAME
002040  001406                          BEQ     18$             ;FN RETURNS NO RESULT
002042  011200                          MOV     @R2,R0          ;GET CURRENT VALUE
002044  001403                          BEQ     19$             ;NO CURRENT VALUE??
002046  142760  000010  000003          BICB    #A.TMP,A.BITS(R0);RESULT IS NOW A TEMP
002054  011512                  19$:    MOV     (SI),@R2        ;RESTORE OLD VALUE
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 52
FNEXIT          EXIT FROM FUNCTION EXECUTION


                                        ;+
                                        ; POP THE F-FRAME AND RETURN TO CALLER WITH RESULT PARSE
                                        ; FRAME ON SI
                                        ;-
002056                                  18$:    POP     R4                      ;GET SUBROUTINE RETURN
002060  014546                                 MOV     -(SI),-(SP)             ;RESTORE CALLER'S RETURN WORD
002062  014546                                 MOV     -(SI),-(SP)             ;RESTORE INPUT CODESTRING PTR
002064  014546                                 MOV     -(SI),-(SP)             ;RESTORE TOKEN BITS
002066  014546                                 MOV     -(SI),-(SP)             ;RESTORE TOKEN VALUE
002070  013705  003314'                       MOV     SIFRAM,SI               ;KILL THE F-FRAME
002074  011500                               MOV     @SI,R0                  ;POINT TO PREVIOUS FRAME
002076  010037  003314'                       MOV     R0,SIFRAM               ;MAKE IT CURRENT FRAME
002102  016037  000002  004764'               MOV     F.FLAG(R0),FLAGS        ;RESTORE PREVIOUS FLAGS
002110  005037  003112'                       CLR     DIMBLK                  ;FUNCTIONS DON'T HAVE DIM ARGS!!
002114  000114                               JMP     @R4                     ;RETURN TO CALLER

002116  104436                  TURK:   TURKEY
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 54
FNEXIT          EXIT FROM FUNCTION EXECUTION

      1         000001                          .END
```

FNEXEC MACRO M1113 08-FEB-80 12:27 PAGE 54-1
SYMBOL TABLE

```
AC0    =%000000          CHRFIL= 000177          CTRMAX  000172R  025 D.CTLM= 000000          EDIFLG  005062R  025
AC1    =%000001          CHRFRO= 000135          C.ALOC= 000162      D.DEAL= 000003          EDIT   = 000003
AC2    =%000002          CHRLBR= 000073          C.BLOK= 000064      D.DECD= 000002          EIS    = 000000
AC3    =%000003          CHRLPR= 000072          C.BLSZ= 000140      D.DIV = 000000          ELESIZ  005132R  025
AC4    =%000004          CHRMNS= 000137          C.BUFF= 177760      D.DMD = 000010          ENDCOR  000004R  025
AC5    =%000005          CHRMUL= 000075          C.BUFL= 177762      D.DQAD= 177777          EOF    = 104475
ALLINT  003274R  025 CHRNEG= 000100          C.CCNT= 177766      D.DROP= 000002          ERASE  = ****** GX
ARYRET  003276R  025 CHRPDL= 000271          C.CPTR= 177764      D.ENCD= 000002          ERRMAX= 000166
ASCQUO= 000042          CHRPLU= 000055          C.CZ   = 001000      D.EQ   = 000000          ERRNUM  004772R  025
ASSIGN= ****** GX       CHRPWR= 000120          C.DIG  = 000040      D.EXPD= 000002          ERRPTR  004770R  025
A.ALEN= 000004          CHRQST= 000121          C.FASW= 020000      D.FGET= 000005          ERR0    004774R  025
A.ARG  = 000040         CHRQUO= 000113          C.FLEN= 000010      D.FMT  = 000004         ESPAWN= ****** GX
A.ASB  = 000200         CHRRBR= 000047          C.FLSH= 004000      D.FPNT= 000005          EVAL1  = ****** GX
A.BITS= 000003          CHRRPR= 000042          C.FRC  = 000001      D.FPUT= 000005          EXEBUF  005054R  025
A.CON  = 000020         CHRSEM= 000074          C.IDEN= 000010      D.GE   = 000000         EXEC   = ****** GX
A.DIMS= 000002          CHRSMO= 000112          C.NAME= 000102      D.GT   = 000000         EXPR   = ****** GX
A.DIM1= 000006          CHRSPA= 000040          C.NOBF= 000400      D.IBM  = 000002         E$CT    005056R  025
A.NUM  = 000006         CHRUPA= 000131          C.NUM  = 000020      D.IND  = 000002         E$PT    005060R  025
A.PTR  = 000000         CHR.BS= 000010          C.OP   = 000004      D.INNR= 000013          FA.APD= 000100
A.TMP  = 000010         CHR.CC= 000003          C.OPN  = 010000      D.INTR= 000006          FA.CRE= 000010
BADDIR= 104463          CHR.CR= 000015          C.RDLN= 177776      D.LE   = 000000         FA.DLK= 001000
BADERR= 104437          CHR.CZ= 000032          C.REWT= 002000      D.LOG  = 000001         FA.ENB= 100000
BLZBUB  005116R  025 CHR.GT= 000046          C.RND  = 000010      D.LT   = 000000         FA.EXC= 002000
BRKCHK= ****** GX       CHR.LF= 000012          C.SBIT= 177756      D.MAX  = 000000         FA.EXT= 000004
BRSELF  005046R  025 CHR.LT= 000043          C.SPFX= 000100      D.MEM  = 000002         FA.NSP= 000100
CCNT    005072R  025 CHR.SI= 000017          C.SPGL= 000200      D.MIN  = 000000         FA.NSR= 000200
CHANEX  003060R  025 CHR.SL= 000057          C.SPLX= 000001      D.MUL  = 000000         FA.POS= 010000
CHAN00  000256R  025 CHR.SO= 000016          C.SPRS= 000002      D.NE   = 000000         FA.RD  = 000001
CHAN01  000330R  025 CHR.TB= 000011          C.SRC  = 177774      D.NN   = 000000         FA.RWD= 004000
CHAN02  000512R  025 CHR.UL= 000106          C.SRCP= 000024      D.NR   = 000000         FA.SEQ= 040000
CHAN03  000674R  025 CHR..A= 000141          C.TT   = 000004      D.OR   = 000000         FA.SHR= 000040
CHAN04  001056R  025 CHR..E= 000145          C.WIDC= 177770      D.PLUS= 000000          FA.TMP= 000020
CHAN05  001240R  025 CHR..S= 000163          C.WIDR= 177772      D.PSUB= 000006          FA.WCK= 020000
CHAN06  001422R  025 CHR..T= 000164          DEBUG  = 000000      D.PWR  = 000001         FA.WRT= 000002
CHAN07  001604R  025 CHR..U= 000165          DEFERR= 104405      D.QQAD= 000000          FD.BLK= 000010
CHAN08  001766R  025 CHR..Z= 000172          DEVIL   005114R  025 D.QUAD= 000000          FD.CCL= 000002
CHAN09  002150R  025 CHR..0= 000060          DEVNTF= 104470      D.RES  = 000000         FD.COM= 020000
CHAN10  002332R  025 CHR..1= 000061          DIGITS  000006R  025 D.RHO  = 000000         FD.CR  = 000002
CHAN11  002514R  025 CH.AND= 000001          DIMAX  = 000020      D.ROT  = 000003         FD.DIR= 000010
CHAN12  002676R  025 CLRSI  = ****** GX      DIMBLK  003112R  025 D.SCON= 000006          FD.FTN= 000001
CHAN13  000022R  023 CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006          FD.F11= 040000
CHAN14  000204R  023 CMDMAX= 000010          DIMTYP  003116R  025 D.SUB  = 000000         FD.INS= 000010
CHNOPN= 104471          CNTSIZ  005130R  025 DIRFUL= 104466      D.SUBS= 000006          FD.ISP= 002000
CHRAND= 000051          COMBIN  005076R  025 DOMERR= 104417      D.TAKE= 000002          FD.MNT= 100000
CHRBEL= 000007          COMCNT  005126R  025 DPTERR= 104416      D.TRAN= 000004          FD.OSP= 004000
CHRBKR= 000133          COMERR= 104425          DYDCNT  003224R  025 D.UNON= 000006          FD.PLC= 000004
CHRBOL= 000200          COMPTR  005122R  025 D.AND  = 000000      D.$MAX= 000000          FD.PRN= 000004
CHRBOX= 000114          COMRTN  003156R  025 D.BXCT= 000007      D.$MIN= 000000          FD.PSE= 010000
CHRCMA= 000054          COMSKP  005124R  025 D.BXIO= 000007      D.$MUL= 000000          FD.RAH= 000001
CHRCOL= 000076          CORPTR  005120R  025 D.BXNL= 000007      D.$PLU= 000000          FD.RAN= 000002
CHRDEL= 000107          CS.ASS= 000200          D.BXPP= 000007      D.$RES= 000000          FD.REC= 000001
CHRDIV= 000053          CS.BOO= 000010          D.BXPW= 000007      D.$SUB= 000000          FD.RWM= 000001
CHRDLR= 000176          CS.FLO= 000020          D.BXRL= 000007      D.1COM= 000001          FD.SDI= 000020
CHRDLT= 000110          CS.IDN= 000040          D.CIRC= 000001      D.1EXP= 000002          FD.SQD= 000040
CHRDOT= 000056          CS.ONE= 000100          D.COMB= 000001      D.1ROT= 000003          FD.TTY= 000004
CHREP  = 000105         CTEPS   000152R  025 D.COMP= 000001      EDICHR  005063R  025 FD.WBH= 000002
CHRERR= 104431          CTMAX   000212R  025 D.CONT= 000006      EDICNT  005064R  025 FF.CHR= 000005
```

SYMBOL TABLE

```
FF.NV = 000003          F.BKDN= 000026          F.R2  = 000014          LENERR= 104412          M.GDUP= 000003
FF.POE= 000002          F.BKDS= 000020          F.R3  = 000016          LENTAB  000054R 025 M.GDWN= 000003
FF.RWD= 000001          F.BKEF= 000050          F.R4  = 000020          LEXPTR  003100R 025 M.GETI= 000006
FF.RWF= 000006          F.BKP1= 000051          F.SEQN= 000100          LFTOP   005066R 025 M.IBM = 000003
FF.SPC= 000004          F.BKST= 000024          F.SPDV= 000072          LINC    005102R 025 M.IDN = 000000
FHEADR  005012R 025     F.BKVB= 000064          F.SPUN= 000074          LINE    005016R 025 M.INSP= 000006
FHRET   005040R 025     F.BR  = 000020          F.STBK= 000036          LININC  005026R 025 M.IOTA= 000000
FILNTF= 104467          F.CHR = 000075          F.STK = 000022          LISIZE= 000020          M.LIT = 000205
FIS   = 000000          F.CNTG= 000034          F.STOP= 000004          LITOP   005072R 025 M.LN  = 000001
FISCON= 046000          F.CODE= 000010          F.STP = 000001          LNLNG = 104541          M.MMD = 000010
FIXPTR  003074R 025     F.DFNB= 000046          F.SUSP= 000002          LOCLCT  005036R 025 M.NEG = 000000
FLAGS   004764R 025     F.DSPT= 000044          F.TRC = 000002          LOWCOR  000002R 025 M.NOT = 000000
FLCBUF  003250R 025     F.DVNM= 000134          F.UNIT= 000136          LSTERP  004766R 025 M.NUB = 000006
FLINE   005014R 025     F.EFBK= 000010          F.URBD= 000020          LWIDTH  000070R 025 M.NUM = 000204
FLOEXP  003270R 025     F.EFN = 000050          F.VAL = 177776          L.BITS= 000003          M.OUTR= 000012
FLONEG  003272R 025     F.END = 002000          F.VBN = 000064          L.CNT = 000002          M.PITM= 000000
FLSWTB  000222R 025     F.EOBB= 000032          F.VBSZ= 000060          L.PRNT= 000001          M.PRDS= 000006
FLTLEN= 000010          F.ERR = 000052          GETNXT= ****** GX       L.PTR = 000000          M.QQAD= 000000
FNBDOM  001470R 020     F.FACC= 000043          GNOME   003102R 025 L.PTR1= 000004          M.QUAD= 000000
FNBR    001116RG 020    F.FFBY= 000014          HASHC   005136R 025 L.SUBS= 000002          M.RAND= 000012
FNBR1   001442R 020     F.FLAG= 000002          HASHTB  005140R 025 MACHIN= 000106          M.RAVL= 000000
FNBR2   001152R 020     F.FLB = 000100          HSHMSK= 177601          MAXCHN= 000014          M.RCIP= 000000
FNBR3   001206R 020     F.FNAM= 000110          HSHSIZ= 000200          MONBLK  003120R 025 M.RDCT= 000011
FNBSYN  001472R 020     F.FNB = 000102          ICHAN   005050R 025 MONCVT  003132R 025 M.REVS= 000003
FNBSYS  001466R 020     F.FNDF= 000010          ILFNAM= 104464          MONLEN  003130R 025 M.ROLL= 000000
FNCALL  000000RG 020    F.FTYP= 000116          ILLCHN= 104540          MONPTR  003122R 025 M.SATA= 000001
FNCAL1  000172R 020     F.FUN = 000010          ILLSW = 104565          MONSFT  003126R 025 M.SCAN= 000011
FNDONE  001106RG 020    F.FVER= 000120          IMONCT= ****** GX       MONTYP  003124R 025 M.SHAP= 000000
FNEXEC  000512RG 020    F.HEAD= 000006          INDERR= 104410          MONVAL  003134R 025 M.SIGN= 000000
FNEXE0  000604R 020     F.HIBK= 000004          INNTMP  003250R 025 M.ABS = 000000          M.SUBR= 000001
FNEXE1  000712R 020     F.LBL = 000020          INNTM2  005016R 025 M.BOL = 000200          M.SYM = 000203
FNEXE6  001064R 020     F.LINE= 000012          INPTR   000064R 025 M.BXAV= 000007          M.SYN = 000201
FNEXIT  001474RG 020    F.LOC = 000004          INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FNIDEN= ****** GX       F.LSTC= 000004          IOERR = 104477          M.BXCT= 000007          M.$ABS= 000000
FNINSI  005044R 025     F.LUN = 000042          I.CNT = 177776          M.BXEX= 000007          M.$CEI= 000000
FORIGN  000032R 025     F.MBCT= 000054          I.DIM = 177772          M.BXFX= 000007          M.$FLR= 000000
FO.APD= 000106          F.MBC1= 000055          I.LEN = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.MFY= 000002          F.MBFG= 000056          I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.RD = 000001          F.MON = 000002          I.OLD = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.UPD= 000006          F.NIL = 000001          I.PTR = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.WRT= 000016          F.NRBD= 000024          I.STR = 177766          M.BXPP= 000007          M.1REV= 000003
FPMODE  000042R 025     F.NREC= 000030          KILL  = ****** GX       M.BXPW= 000007          M.1SCN= 000011
FPP   = 000000          F.OPN = 000010          LABERR= 104406          M.BXRL= 000007          NAMBUF  003240R 025
FPPLEN= 000004          F.OVBS= 000030          LABLPT  005042R 025 M.BXTS= 000007          NB.DEV= 000200
FREBLK  007400R 025     F.PREV= 000000          LARGBL  003162R 025 M.BXWA= 000007          NB.DIR= 000100
FUZZ    000142R 025     F.PRNT= 100000          LARGCV  003176R 025 M.CEI = 177777          NB.NAM= 000004
FXPTR1  003076R 025     F.PROT= 010000          LARGFL  003160R 025 M.CHR = 000202          NB.SD1= 000400
F$EVAL= 000100          F.RACC= 000016          LARGLE  003174R 025 M.CMT = 000206          NB.SD2= 001000
F$EVI = 000200          F.RATT= 000001          LARGLN  003166R 025 M.DOUT= 000006          NB.SNM= 000040
F$EXEC= 000400          F.RCNM= 000034          LARGPT  003164R 025 M.DQAD= 000006          NB.STP= 000020
F$FUNC= 001000          F.RCTL= 000017          LARGSF  003172R 025 M.ENCD= 000004          NB.SVR= 000010
F.ACTL= 000076          F.READ= 004000          LARGTY  003170R 025 M.EXE = 000000          NB.TYP= 000002
F.ALOC= 000040          F.REDF= 000040          LARGVA  003200R 025 M.EXP = 000001          NB.VER= 000001
F.BBFS= 000062          F.RES = 000040          LBINC   005100R 025 M.FACT= 000001          ND.DEV= 000200
F.BDB = 000070          F.RET = 177772          LBUFF   003316R 025 M.FIL = 000177          ND.EXT= 000002
F.BGBC= 000057          F.RSIZ= 000002          LBUFLN= 000621          M.FLR = 177777          ND.NAM= 000004
F.BITS= 177774          F.RTYP= 000000          LCOUNT  005106R 025 M.FMT = 000004          ND.PPN= 000100
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 54-3
SYMBOL TABLE

ND.$EX= 000000          OPTMP0  005066R     025 RESVAL  003154R     025 SYMPT1  005346R     025 TURKEY= 104436
ND.$E$= 000020          OPTMP1  005070R     025 RETCOR= ****** GX       SYMTAB  005352R     025 TYPE  = 000166
ND.$NM= 000000          OPTMP2  005072R     025 RINC    005104R     025 SYMTBF= 104404          T.BOL = 020000
ND.$N$= 000040          OPTMP3  005074R     025 RLTBAS  005066R     025 SYMTOP  005350R     025 T.CON = 004000
ND.$PN= 001400          OPTMP4  005076R     025 ROTINC  005070R     025 SYMTP   006272R     025 T.DIM = 000100
NEGSGN  005063R   025   OPTMP5  005100R     025 ROUND   000010R     025 SYMWRN= 000054          T.FUN = 010000
NEWLIN= ****** GX       OPTMP6  005102R     025 ROVER   003312R     025 SYNERR= 104407          T.LIST= 000400
NINPT1  003302R   025   OPTMP7  005104R     025 RSTS  = 000000          SYSER = ****** GX     T.LIT = 000020
NINPT2  003300R   025   OPTMP8  005106R     025 RSX   = 000115          SYSERR= 104402          T.MON = 000010
NINSTK  003304R   025   OPTMP9  005110R     025 RTOP    005070R     025 SYSTAT  000000R     024 T.NIL = 000004
NINTMP  003310R   025   OPTM10  005112R     025 RT11  = 000000          SY.BAD= 000001          T.NUM = 002000
NINTYP  003306R   025   OPTM14  005026R     025 R.BITS= 000003          SY.BRK= 100000          T.OPFN= 000001
NOTAVL= 104472          ORIGIN  000030R     025 R.FIX = 000001          SY.BS = 000010          T.OTHR= 040000
NOTOPN= 104473          OUTCHR= ****** GX       R.LCT = 000002          SY.CLR= 000040          T.SOP = 000002
NPRT1   004762R   025   OUTPTR  000066R     025 R.LN  = 000002          SY.ERP= 000020          T.SPOP= 000040
NULL    007364R   025   OVROPR= 000000          R.LNUM= 000010          SY.NOU= 000100          T.TMP = 000200
NUMCHN= 000020          PATCH   007344R     025 R.NEXT= 000006          SY.CPN= 000002          T.VAL = 100000
NUMDFL  003230R   025   PKILL = ****** GX       R.OFF = 000000          SY.PMT= 001000          T.VAR = 001000
NUMEFL  003233R   025   PRECSN  000052R     025 R.PREV= 000004          SY.SRT= 000400          UPPER = 177400
NUMESN  003236R   025   PRINT = ****** GX       R.PTR = 000000          SY.SVC= 000004          UTNAME  000344R     023
NUMFLG  003230R   025   RANDAC  003064R     025 R.SEQ = 000003          SY.SVL= 000200          VALERR= 104413
NUMLEN  000062R   025   RANDY   003062R     025 R.VAR = 000002          S.AB  = 177776          VARBLE  007400R     025
NUMNEG  003232R   025   RANERR= 104411          R6    =%000006          S.ASB = 100000          VARBLS  000000R     025
NUMQFL  003234R   025   RARGBL  003204R     025 R7    =%000007          S.BFHD= 000020          VERNUM  000000R     025
NUMRIG  003231R   025   RARGCV  003220R     025 SI    =%000005          S.BITS= 000002          VRSION= 000002
NUMWID  003226R   025   RARGFL  003202R     025 SIDAMN= 104433          S.CHR1= 000003          WIDTH   000072R     025
NUMZFL  003235R   025   RARGLE  003216R     025 SIFRAM  003314R     025 S.FATT= 000016          WROUND  000020R     025
NYIERR= 104432          RARGLN  003210R     025 SISTAK  006350R     025 S.FDB = 000140          WSDATE  000136R     025
N.BLSZ= 000036          RARGPT  003206R     025 SITOP   006346R     025 S.FLB = 040000          WSFULL= 104403
N.DESC= 000020          RARGSF  003214R     025 SITP    007144R     025 S.FNAM= 000006          WSNAME  000074R     025
N.DEV = 000032          RARGTY  003212R     025 SIWARN= 000200          S.FNB = 000036          WSTIME  000132R     025
N.DEVN= 000034          RARGVA  003222R     025 SIZTAB  000044R     025 S.FNBW= 000017          $FIS  = ****** G
N.DID = 000024          RBASE   005074R     025 SI.BIT= 177774          S.FNTY= 000004          $FPP  = ****** G
N.DVNM= 000032          RCOUNR  005112R     025 SI.LEN= 177772          S.FTYP= 000002          $FPPLN= ****** G
N.EXT = 000014          RCOUNT  005110R     025 SI.RET= 177772          S.LEN = 000002          $MACHN= ****** G
N.FID = 000000          REDCNT  005070R     025 SI.VAL= 177776          S.LINK= 177774          $RSTS = ****** G
N.FNAM= 000006          REDIGN  005066R     025 SKPSIZ  005134R     025 S.NFEN= 000020          $RSX  = ****** G
N.FTYP= 000014          REDOFF  005072R     025 SPHOLD  000004R     024 S.TYPE= 140000          $RT11 = ****** G
N.FVER= 000016          REENTR  003060R     025 SPINIT  000002R     024 S.VAL = 000000          .BOOLE= 000002
N.NAME= 000006          RESBIT  003142R     025 STKINC= 000774          TKDIMS  005072R     025 .CHARA= 000000
N.NEXT= 000022          RESCVT  003152R     025 STKXTN= ****** GX       TKFRAM  005066R     025 .DEF. = 000002
N.PPN = 000024          RESERR= 104440          STOP  = 104420          TKLEFT  005070R     025 .FLOAT= 000006
N.STAT= 000020          RESLEN  003150R     025 SUBCNT  003104R     025 TMODE   000403R     023 .FSRCB= ****** G
N.UNIT= 000034          RESPTR  003140R     025 SUBDCT  003106R     025 TMPTAG= 000166          .INTEG= 000004
OBUFLN= 001000          RESSFT  003146R     025 SUBPTR  003110R     025 TTYBUF  004140R     025 .MOLUN  000404R     023
OCHAN   005052R   025   RESTYP  003144R     025 SYMB1   005344R     025 TTYLEN= 000622          ...GBL= 000000
OMODE   000402R   023   RESULT  003136R     025 SYMF    005340R     025 TURK    002116R     020 ...TPC= 000020

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
```

```
FNEXEC  MACRO M1113  08-FEB-80 12:27  PAGE 54-4
SYMBOL TABLE

CHRTBL  000000      011
VTBL    000000      012
ASCTBL  000000      013
OVRTBL  000000      014
TRNTBL  000000      015
MNTBL   000000      016
MNTRTL  000000      017
CODE    002120      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13320 WORDS  ( 53 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:36
FNEXEC,DM1:[305,301]FNEXEC/-SP=APLRX1,APLDEF,FNEXEC,END
```

FILEID**FNHEAD

```
FFFFFFFFFF    NN       NN    HH       HH    EEEEEEEEEE      AAAAAA      DDDDDDDD
FFFFFFFFFF    NN       NN    HH       HH    EEEEEEEEEE      AAAAAA      DDDDDDDD
FF            NN       NN    HH       HH    EE             AA    AA    DD      DD
FF            NN       NN    HH       HH    EE             AA    AA    DD      DD
FF            NNNN     NN    HH       HH    EE             AA    AA    DD      DD
FF            NNNN     NN    HH       HH    EE             AA    AA    DD      DD
FFFFFFFF      NN  NN   NN    HHHHHHHHHH    EEEEEEEE       AA    AA    DD      DD
FFFFFFFF      NN  NN   NN    HHHHHHHHHH    EEEEEEEE       AA    AA    DD      DD
FF            NN    NNNN    HH       HH    EE             AAAAAAAAAA    DD      DD
FF            NN    NNNN    HH       HH    EE             AAAAAAAAAA    DD      DD
FF            NN      NN    HH       HH    EE             AA    AA    DD      DD
FF            NN      NN    HH       HH    EE             AA    AA    DD      DD
FF            NN      NN    HH       HH    EEEEEEEEEE     AA    AA    DDDDDDDD
FF            NN      NN    HH       HH    EEEEEEEEEE     AA    AA    DDDDDDDD


LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

.MAIN. MACRO M1113 08-FEB-80 12:30
TABLE OF CONTENTS


```
     2-    1    APLDEF
     2-    8    A P L
     2-    9
     2-   10    A PROGRAMMING LANGUAGE
     2-   11
     2-   12    STEVE POULSEN & DAVE VANN
     2-   13    11-Jul-76
     2-   14    ROBERT GOTTLIEB & PETER HAYNES
     2-   15    2-Feb-79
     2-   16
     2-   17    V2
     2-   18
     2-   19
     2-   20    COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980
     2-   21    DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
     2-   22
     2-   23
     2-   24    THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY  ON  A  SINGLE
     2-   25    COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
     2-   26    COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
     2-   27    BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
     2-   28    USE ON SUCH SYSTEM AND TO ONE  WHO  AGREES  TO  THESE  LICENSE  TERMS.
     2-   29    TITLE  TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN
     2-   30    DEC.
     2-   31
     2-   32    THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
     2-   33    AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
     2-   34    CORPORATION.
     2-   35
     2-   36    DEC ASSUMES NO RESPONSIBILITY  FOR  THE  USE  OR  RELIABILITY  OF  ITS
     2-   37    SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
     2-   38
     2-   39
     3-   46    DEFINITIONS      ASSEMBLY PARAMETERS
     4-  139    DEFINITIONS      OPERATOR OVERLAYS
     5-  268    DEFINITIONS      REGISTERS
     6-  331    DEFINITIONS      MEMORY ORGANIZATION
     7-  377    DEFINITIONS      ORDERING PSECTS
     8-  426    DEFINITIONS      SI PARSE FRAMES
     9-  474    DEFINITIONS      TOKEN DESCRIPTION BITS
    10-  515    DEFINITIONS      SI EXECUTION FRAMES
    11-  580    DEFINITIONS      SPECIAL PARSE CHARACTERS
    12-  607    DEFINITIONS      CHRTAB BITS
    12-  624    DEFINITIONS      SYSTEM STATUS BITS
    13-  640    DEFINITIONS      F-FRAMES
    14-  724    DEFINITIONS      FUNCTION LINE BLOCKS
    16-  807    DEFINITIONS      ARRAY STORAGE BLOCK
    17-  869    DEFINITIONS      LIST STORAGE BLOCK
    18-  911    DEFINITIONS      INDEXING STACK FRAME
    19-  940    DEFINITIONS      SYMBOL TABLE ENTRY
    20-  969    DEFINITIONS      CHANNEL BLOCK
    22- 1085    DEFINITIONS              RSX DATA
    23- 1110    DEFINITIONS      NAME BLOCKS
    33- 1612    DEFINITIONS      MACROS
    35- 1905    DEFINITIONS      APL CHARACTERS
    36- 1967    DEFINITIONS      ERRORS
    44-    4    FNHEAD           ANALYZE FUNCTION HEADER
```

```
FNIHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 1


      1        000000                  RT11    =  0          ;NOT RT-11
      2        000000                  RSTS    =  0          ;NOT RSTS/E
      3        000115                  RSX     = 'M          ;RSX-11/M
      4
      5        000000                  EIS     =  0          ;NO EIS
      6        000000                  FIS     =  0          ;NO FIS
      7        000000                  FPP     =  0          ;NO FPP
      8
      9        000004                  FPPLEN  =  4.         ;SINGLE PRECISION
     10
     11        000000                  DEBUG   =  0          ;DON'T DISPLAY REGISTERS ON ERRORS
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 2
APLDEF

                                .SBTTL  APLDEF
                                .IDENT  \V02.03\
```

```
APL     MACRO M1113  08-FEB-80 12:30  PAGE 42
DEFINITIONS     ERRORS

   2513                                 .LIST  SEQ,SRC,LOC,BIN,BEX,MEB
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 44
DEFINITIONS     ERRORS

                                        .TITLE  FNHEAD
                                        .IDENT  \V02.03\

                                        .SBTTL  FNHEAD          ANALYZE FUNCTION HEADER

                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       'FNHEAD' CHECK THE SYNTAX FOR AND SETS UP THE POINTERS
                                        ;       SO THAT THEY CAN BE COPPIED INTO A FUNCTION HEADER BLOCK.
                                        ;       WHEN THIS ROUTINE RETURNS, THE SYMBOL TABLE     ENTRY POINTERS
                                        ;       (POINTERS TO THE SYMBOL BLOCKS) FOR THE RESULT, ARGS,
                                        ;       FUNCTION NAME, AND LOCAL VARIABLES WILL BE ON THE SP STACK.
                                        ; INPUTS:
                                        ;       R0 -    POINTS TO FIRST CHAR PRODUCED BY 'FIXLIN'
                                        ;       R1 -    CONTAINS BYTE COUNT OF CHARS PRODUCED BY 'FIXLIN'
                                        ;--
 000000                                         .PSECT  CODE

 000000                                 FNHEAD::
 000000                                         PUSH    SI                      ;Save the SI Value
 000002  010637  005040'                        MOV     SP,FHRET                ;Save the stack Pointer
 000006                                         CALL    LEX                     ;ANALYZE FN HEADER
 000012  005037  003136'                        CLR     RESULT                  ;CLEAR RESULT FLAG
 000016  005037  005036'                        CLR     LOCLCT                  ;NO LOCAL VARIABLES YET
 000022  005037  003142'                1$:     CLR     RESBIT                  ;USED FOR ARG COUNTING
 000026                                         CALL    GETNXT                  ;GET NEXT TOKEN (RIGHT TO LEFT!)
 000032  032703  011000                2$:     BIT     #T.VAR+T.FUN,R3         ;IS IT A SYMBOL?
 000036  001435                                BEQ     7$                      ;NO, WE MUST BE DONE
 000040                                         PUSH    R4                      ;PUSH POINTER TO SYMBOL BLOCK
 000042  005737  003142'                        TST     RESBIT                  ;FOUND ANY ARGS YET?
 000046  001002                                BNE     3$                      ;YES, THEN THIS IS ALSO ARG
 000050  005237  005036'                        INC     LOCLCT                  ;ELSE THIS IS PROBABLY A LOCAL VAR.
 000054  005237  003142'                3$:     INC     RESBIT                  ;COUNT IT AS AN ARG
 000060  023727  003142'  000003               CMP     RESBIT,#3               ;TOO MANY ARGS YET?
 000066  003402                                BLE     5$                      ;NO, STILL OK
 000070  000137  000654'                4$:     JMP     23$                     ;Error, do error return
 000074                                 5$:     CALL    GETNXT                  ;GET NEXT TOKEN
 000100  023727  003142'  000001               CMP     RESBIT,#1               ;ONLY 1 ARG SO FAR?
 000106  003351                                BGT     2$                      ;NO, CHECK IF THIS IS ONE
 000110  032703  040000                        BIT     #T.OTHR,R3              ;YES, PERHAPS THIS IS A LOCAL VAR?
 000114  001403                                BEQ     6$                      ;NO, THEN WE MUST ASSUME FIRST ARG
 000116  122704  000074                        CMPB    #CHRSEM,R4              ;IS CHAR A SEMI-COLON?
 000122  001737                                BEQ     1$                      ;YES! IT WAS A LOCAL VAR, NO ARGS YET!
 000124  005337  005036'                6$:     DEC     LOCLCT                  ;NOT A LOCAL VAR SO CORRECT COUNT
 000130  000740                                BR      2$                      ;AND CHECK FOR ARG
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 45
FNHEAD           ANALYZE FUNCTION HEADER


                                        ;+
                                        ; NOW WE KNOW HOW MANY LOCAL VARIABLES THERE ARE ('LOCLCT')
                                        ; AND THEY ARE ON THE STACK FOLLOWED BY ALL OF THE ARGS
                                        ; (COUNTED IN 'RESBIT').  NOW WE MUST CHECK TO SEE IF A
                                        ; BACK ARROW IS PRESENT INDICATING THAT THE FUNCTION RETURNS
                                        ; A RESULT, OR IF A DEL IS FOUND IN WHICH CASE WE ARE DONE.
                                        ;-
000132  032703  020000          7$:     BIT     #T.BOL,R3       ;END OF LINE?
000136  001044                          BNE     10$             ;YES, FINISH UP AND EXIT
000140  032703  040000                  BIT     #T.OTHR,R3      ;IS TOKEN A CHAR?
000144  001751                          BEQ     4$              ;NOPE, DEFN ERROR
0(0146  122704  000133                  CMPB    #CHRBKR,R4      ;BACK ARROW?
0(0152  001020                          BNE     8$              ;No, finished
                                        ;+
                                        ; WE HAVE A BACKARROW, SO FUNCTION RETURNS RESULTS.  GET
                                        ; SYMBOL FOR RESULT, SET FLAG ('RESULT') AND FINISH UP.
                                        ;-
000154                                  CALL    GETNXT          ;GET NEXT TOKEN
000160  032703  011000                  BIT     #T.VAR+T.FUN,R3 ;SYMBOL?
000164  001741                          BEQ     4$              ;NO, DEFN ERROR
000166                                  PUSH    R4              ;PUSH   POINTER TO RESULT SYMBOL
000170  005237  0031C                   INC     RESULT          ;FN RETURNS RESULT
000174                                  CALL    GETNXT          ;GET NEXT TOKEN
000200  032703  020000                  BIT     #T.BOL,R3       ;END OF LINE?
000204  001021                          BNE     10$             ;YES, ALL DONE
000206  032703  040000                  BIT     #T.OTHR,R3      ;CHARACTER?
000212  001726                          BEQ     4$              ;NO, ERROR (SHOULD BE DEL!)
000214  122704  000107          8$:     CMPB    #CHRDEL,R4      ;IS CHAR A DEL?
000220  001406                          BEQ     9$              ;YES, IS GOOD
000222  122704  000271                  CMPB    #CHRPDL,R4      ;A PROTECT DEL?
000226  001320                          BNE     4$              ;NO, THEN SYNTAX IS BAD
000230  052737  010000  004764'        BIS     #F.PROT,FLAGS   ;MUSN'T  FORGET
000236                          9$:     CALL    GETNXT          ;GET NEXT TOKEN
000242  032703  020000                  BIT     #T.BOL,R3       ;ARE WE DONE WITH LINE?
000246  001566                          BEQ     20$             ;NO, THEN THERE IS GARBAGE IN LINE
000250  013701  003142'          10$:    MOV     RESBIT,R1       ;GET ARG COUNT
000254  005737  003136'                  TST     RESULT          ;RETURN A RESULT?
000260  001402                          BEQ     11$             ;NOPE
000262  062701  000004                  ADD     #4,R1           ;YES, MARK IT

000266  010137  003142'          11$:    MOV     R1,RESBIT       ;SAVE FUNCTION DESCRIPTION BITS
000272  001554                          BEQ     20$             ;Zero, Definition error
000274  032701  177770                  BIT     #^C<7>,R1       ;ANY EXTRA BITS?
000300  001151                          BNE     20$             ;YES, TOO MANY ARGS PERHAPS??
000302  017705  005040'                 MOV     @FHRET,SI       ;Restore SI Pointer
000306  013700  005036'                 MOV     LOCLCT,R0       ;GET COUNT OF LOCAL VARIABLES
000312  006300                          ASL     R0              ;SHIFT TO GET WORD COUNT
000314  062700  000020                  ADD     #R.LNUM+10,R0   ;ADD HEADER BLOCK SIZE
000320  063700  000062'                 ADD     NUMLEN,R0       ;ADD LINE NUMBER SIZE
000324                                  CALL    GETFRA          ;GET ROOM FOR NEW FUNCTION HEADER
000330  010004                          MOV     R0,R4           ;COPY POINTER TO FN HEADER
000332  005020                          CLR     (R0)+           ;WE DON'T KNOW ITS NAME YET
000334  113701  005036'                 MOVB    LOCLCT,R1       ;GET COUNT OF LOCAL VARIABLES
000340  110120                          MOVB    R1,(R0)+        ;SAVE LOCAL COUNT IN HEADER
000342  112720  000140                  MOVB    #F.FLB+F.RES,(R0)+ ;INITIALLY ASSUME FN RETURNS RESULT
000346  010420                          MOV     R4,(R0)+        ;PREVIOUS LINE IS US
000350  010420                          MOV     R4,(R0)+        ;NEXT FUNCTION LINE IS US (NO LINES YET)
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 45-1
FNHEAD           ANALYZE FUNCTION HEADER

    104 000352                                          CLRF    (R0)+           ;OUR LINE NUMBER IS 0.0  (HEADER)
Z       000356  010020                                  MOV R0,(R0)+
    105 000360  005020                                  CLR     (R0)+           ;NO LABELS YET
    106 000362  013703  003142'                         MOV     RESBIT,R3       ;GET FN DESCRIPTION BITS
    107 000366  032703  000004                          BIT     #4,R3           ;FN RETURN A VALUE??
    108 000372  001004                                  BNE     12$             ;YES, SAVE POINTER TO VALUE SYMBOL
    109 000374  005046                                  CLR     -(SP)           ;NO, PUSH A ZERO
    110 000376  142764  000040  000003                  BICB    #F.RES,R.BITS(R4);MARK AS NOT RETURNING A VALUE
    111 000404  012620                          12$:    MOV     (SP)+,(R0)+     ;COPY RESULT POINTER (0=NONE)
    112 000406  042703  000004                          BIC     #4,R3           ;KILL DESCRIPTION BIT
    113 000412  001520                                  BEQ     23$             ;OOPS! ILLEGAL FUNCTION DEF
    114 000414  022703  000003                          CMP     #3,R3           ;IS FN DYADIC (2 ARGS?)
    115 000420  001404                                  BEQ     13$             ;YES, POP LEFT ARG'S NAME
    116 000422  005046                                  CLR     -(SP)           ;NO, PUSH A ZERO
    117 000424  152764  000002  000003                  BISB    #F.MON,R.BITS(R4);GUESS THAT THE FN IS MONADIC
    118 000432  012620                          13$:    MOV     (SP)+,(R0)+     ;COPY LEFT ARG PTR INTO HEADER
    119 000434  012614                                  MOV     (SP)+,@R4       ;POP FN NAME INTO HEADER
    120 000436  032703  000002                          BIT     #2,R3           ;IS FN REALLY NILADIC?
    121 000442  001007                                  BNE     14$             ;NOPE, GET RIGHT ARG
    122 000444  005046                                  CLR     -(SP)           ;YUP, PUSH A ZERO
    123 000446  142764  000002  000003                  BICB    #F.MON,R.BITS(R4);FUNCTION ISN'T REALLY MONADIC
    124 000454  152764  000001  000003                  BISB    #F.NIL,R.BITS(R4);IT'S NILADIC!
    125 000462  012620                          14$:    MOV     (SP)+,(R0)+     ;COPY RIGHT ARG INTO FN HEADER
    126 000464  013702  005036'                         MOV     LOCLCT,R2       ;GET COUNT OF LOCAL VARIABLES
    127 000470  001403                                  BEQ     16$             ;IF THERE ARE ANY
    128 000472  012620                          15$:    MOV     (SP)+,(R0)+     ;STORE NEXT LOCAL PTR IN HEADER
    129 000474                                          SOB     R2,15$          ;COPY ALL LOCAL VARIABLES INTO HEADER
    130 000500  011402                          16$:    MOV     @R4,R2          ;GET FN NAME POINTER INTO SYMTAB
    131 000502  032737  000040  004764'                 BIT     #F.REDF,FLAGS   ;IS THIS A HEADER REDEFINITION?
    132 000510  001436                                  BEQ     19$             ;NO, GOOD
    133 000512  042737  000040  004764'                 BIC     #F.REDF,FLAGS   ;CLEAR THE FLAG
    134 000520  011200                                  MOV     (R2),R0         ;Get pointer to old header
    135 000522  001003                                  BNE     17$             ;Yes, there is one
    136 000524  013700  005012'                         MOV     FHEADR,R0       ;Point to old fn header
    137 000530  001426                                  BEQ     19$             ;If there is one
    138 000532  016001  000004                  17$:    MOV     R.PREV(R0),R1   ;GET LAST LINE
    139 000536  020100                                  CMP     R1,R0           ;EMPTY FUNCTION?
    140 000540  001412                                  BEQ     18$             ;YES
    141 000542  010164  000004                          MOV     R1,R.PREV(R4)   ;MAKE LAST LINE OF NEW HEADER
    142 000546  010461  000006                          MOV     R4,R.NEXT(R1)   ;SET PTR TO NEW HEADER IN LINE
    143 000552  016001  000006                          MOV     R.NEXT(R0),R1   ;POINT TO FIRST FN LINE
    144 000556  010164  000006                          MOV     R1,R.NEXT(R4)   ;IT IS NEW FN'S FIRST LINE
    145 000562  010461  000004                          MOV     R4,R.PREV(R1)   ;FIX UP BACK POINTER IN FIRST LINE
    146 000566  152764  000010  000003          [illegible]  #F.OPN,R.BITS(R4)  ;Yes, it is really open
    147 000574  005070  000000                          CLR     @0(R0)          ;KILL SYMBOL PTR TO HEADER
    148 000600                                          CALL    RETCOR          ;KILL OLD FUNCTION HEADER
    149 000604  000407                                  BR      20$             ;AND LET NEW HEADER TAKE OVER
    150 000606  005712                          19$:    TST     @R2             ;IS NAME ALREADY IN USE?
    151 000610  001405                                  BEQ     20$             ;No, is OK
    152 000612  016500  177776                          MOV     SI.VAL(SI),R0   ;Point to Function Header
    153 000616                                          CALL    RETCOR          ;Kill the Block
    154 000622  000407                                  BR      21$             ;Error Return
    155 000624  013700  005040'                 20$:    MOV     FHRET,R0        ;Stack Pointer
    156 000630  062760  000002  000002                  ADD     #2.,2(R0)       ;Reset return address
    157 000636  010400                                  MOV     R4,R0           ;COPY POINTER TO HEADER
    158 000640  010412                                  MOV     R4,@R2          ;MAKE NAME POINT TO FUNCTION HEADER
    159 000642  000241                          21$:    CLC                     ;Normal return
```

```
    160 000644  013706  005040'         22$:    MOV     FHRET,SP        ;Restore the Stack
    161 000650                                  POP     SI              ;Restore SI value
    162 000652                                  RETURN                  ;   and return
    163
    164 000654  000261                  23$:    SEC                     ;Error return
    165 000656  000772                          BR      22$             ;And do it
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 47
FNHEAD           ANALYZE FUNCTION HEADER

      1          000001                              .END
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 47-1
SYMBOL TABLE

AC0    =%000000          CHRLBR= 000073    C.BLSZ= 000140    D.DIV = 000000    ELESIZ  005132R  025
AC1    =%000001          CHRLPR= 000072    C.BUFF= 177760    D.DMD = 000010    ENDCOR  000004R  025
AC2    =%000002          CHRMNS= 000137    C.BUFL= 177762    D.DQAD= 177777    EOF    = 104475
AC3    =%000003          CHRMUL= 000075    C.CCNT= 177766    D.DROP= 000002    ERRMAX= 000166
AC4    =%000004          CHRNEG= 000100    C.CPTR= 177764    D.ENCD= 000002    ERRNUM  004772R  025
AC5    =%000005          CHRPDL= 000271    C.CZ   = 001000   D.EQ   = 000000    ERRPTR  004770R  025
ALLINT  003274R  025      CHRPLU= 000055    C.DIG  = 000040   D.EXPD= 000002    ERRO    004774R  025
ARYRET  003276R  025      CHRPWR= 000120    C.FASW= 020000    D.FGET= 000005    EXEBUF  005054R  025
ASCQUO= 000042            CHRQST= 000121    C.FLEN= 000010    D.FMT  = 000004   E$CT    005056R  025
A.ALEN= 000004            CHRQUO= 000113    C.FLSH= 004000    D.FPNT= 000005    E$PT    005060R  025
A.ARG  = 000040           CHRRBR= 000047    C.FRC  = 000001   D.FPUT= 000005    FA.APD= 000100
A.ASB  = 000200           CHRRPR= 000042    C.IDEN= 000010    D.GE   = 000000   FA.CRE= 000010
A.BITS= 000003            CHRSEM= 000074    C.NAME= 000102    D.GT   = 000000   FA.DLK= 001000
A.CON  = 000020           CHRSMO= 000112    C.NOBF= 000400    D.IBM  = 000002   FA.ENB= 100000
A.DIMS= 000002            CHRSPA= 000040    C.NUM  = 000020   D.IND  = 000002   FA.EXC= 002000
A.DIM1= 000006            CHRUPA= 000131    C.OP   = 000004   D.INNR= 000013   FA.EXT= 000004
A.NUM  = 000006           CHR.BS= 000010    C.OPN  = 010000   D.INTR= 000006   FA.NSP= 000100
A.PTR  = 000000           CHR.CC= 000003    C.RDLN= 177776    D.LE   = 000000   FA.NSR= 000200
A.TMP  = 000010           CHR.CR= 000015    C.REWT= 002000    D.LOG  = 000001   FA.POS= 010000
BADDIR= 104463            CHR.CZ= 000032    C.RND  = 000010   D.LT   = 000000   FA.RD  = 000001
BADERR= 104437            CHR.GT= 000046    C.SBIT= 177756    D.MAX  = 000000   FA.RWD= 004000
BLZBUB  005116R  025      CHR.LF= 000012    C.SPFX= 000100    D.MEM  = 000002   FA.SEQ= 040000
BRSELF  005046R  025      CHR.LT= 000043    C.SPGL= 000200    D.MIN  = 000000   FA.SHR= 000040
CCNT    005072R  025      CHR.SI= 000017    C.SPLX= 000001    D.MUL  = 000000   FA.TMP= 000020
CHANEX  003060R  025      CHR.SL= 000057    C.SPRS= 000002    D.NE   = 000000   FA.WCK= 020000
CHAN00  000256R  025      CHR.SO= 000016    C.SRC  = 177774   D.NN   = 000000   FA.WRT= 000002
CHAN01  000330R  025      CHR.TB= 000011    C.SRCP= 000024    D.NR   = 000000   FD.BLK= 000010
CHAN02  000512R  025      CHR.UL= 000106    C.TT   = 000004   D.OR   = 000000   FD.CCL= 000002
CHAN03  000674R  025      CHR..A= 000141    C.WIDC= 177770    D.PLUS= 000000    FD.COM= 020000
CHAN04  001056R  025      CHR..E= 000145    C.WIDR= 177772    D.PSUB= 000006    FD.CR  = 000002
CHAN05  001240R  025      CHR..S= 000163    DEBUG  = 000000   D.PWR  = 000001   FD.DIR= 000010
CHAN06  001422R  025      CHR..T= 000164    DEFERR= 104405    D.QQAD= 000000    FD.FTN= 000001
CHAN07  001604R  025      CHR..U= 000165    DEVIL   005114R  025 D.QUAD= 000000  FD.F11= 040000
CHAN08  001766R  025      CHR..Z= 000172    DEVNTF= 104470    D.RES  = 000000   FD.INS= 000010
CHAN09  002150R  025      CHR..0= 000060    DIGITS  000006R  025 D.RHO  = 000000  FD.ISP= 002000
CHAN10  002332R  025      CHR..1= 000061    DIMAX  = 000020   D.ROT  = 000003   FD.MNT= 100000
CHAN11  002514R  025      CH.AND= 000001    DIMBLK  003112R  025 D.SCON= 000006  FD.OSP= 004000
CHAN12  002676R  025      CMDBUF  003250R  025 DIMPTR  003114R  025 D.SDIF= 000006  FD.PLC= 000004
CHAN13  000022R  023      CMDMAX= 000010    DIMTYP  003116R  025 D.SUB  = 000000  FD.PRN= 000004
CHAN14  000204R  023      CNTSIZ  005130R  025 DIRFUL= 104466    D.SUBS= 000006   FD.PSE= 010000
CHNOPN= 104471            COMBIN  005076R  025 DOMERR= 104417    D.TAKE= 000002   FD.RAH= 000001
CHRAND= 000051            COMCNT  005126R  025 DPTERR= 104416    D.TRAN= 000004   FD.RAN= 000002
CHRBEL= 000007            COMERR= 104425    DYDCNT  003224R  025 D.UNON= 000006  FD.REC= 000001
CHRBKR= 000133            COMPTR  005122R  025 D.AND  = 000000   D.$MAX= 000000   FD.RWM= 000001
CHRBOL= 000200            COMRTN  003156R  025 D.BXCT= 000007    D.$MIN= 000000   FD.SDI= 000020
CHRBOX= 000114            COMSKP  005124R  025 D.BXIO= 000007    D.$MUL= 000000   FD.SQD= 000040
CHRCMA= 000054            CORPTR  005120R  025 D.BXNL= 000007    D.$PLU= 000000   FD.TTY= 000004
CHRCOL= 000076            CS.ASS= 000200    D.BXPP= 000007    D.$RES= 000000   FD.WBH= 000002
CHRDEL= 000107            CS.BCO= 000010    D.BXPW= 000007    D.$SUB= 000000   FF.CHR= 000005
CHRDIV= 000053            CS.FLO= 000020    D.BXRL= 000007    D.1COM= 000001   FF.NV  = 000003
CHRDLR= 000176            CS.IDN= 000040    D.CIRC= 000001    D.1EXP= 000002   FF.POE= 000002
CHRDLT= 000110            CS.ONE= 000100    D.COMB= 000001    D.1ROT= 000003   FF.RWD= 000001
CHRDOT= 000056            CTEPS   000152R  025 D.COMP= 000001    EDICHR  005063R  025 FF.RWF= 000006
CHREP  = 000105           CTMAX   000212R  025 D.CONT= 000006    EDICNT  005064R  025 FF.SPC= 000004
CHRERR= 104431            CTRMAX  000172R  025 D.CTLM= 000000    EDIFLG  005062R  025 FHLEN   005012R  025
CHRFIL= 000177            C.ALOC= 000162    D.DEAL= 000003    EDIT   = 000003   FHRET   005040R  025
CHRFRO= 000135            C.BLOK= 000064    D.DECD= 000002    EIS    = 000000   FILNTF= 104467
```

FNHEAD MACRO M1113 08-FEB-80 12:30 PAGE 47-2
SYMBOL TABLE

```
FIS    = 000000          F.FNB  = 000102          HSHSIZ= 000200          MONCVT  003132R     025 M.REVS= 000003
FISCON= 046000           F.FNDF= 000010          ICHAN   005050R     025 MONLEN  003130R     025 M.ROLL= 000000
FIXPTR  003074R     025 F.FTYP= 000116          ILFNAM= 104464          MONPTR  003122R     025 M.SATA= 000001
FLAGS   004764R     025 F.FUN  = 000010          ILLCHN= 104540          MONSFT  003126R     025 M.SCAN= 000011
FLCBUF  003250R     025 F.FVER= 000120          ILLSW  = 104565          MONTYP  003124R     025 M.SHAP= 000000
FLINE   005014R     025 F.HEAD= 000006          INDERR= 104410          MONVAL  003134R     025 M.SIGN= 000000
FLOEXP  003270R     025 F.HIBK= 000004          INNTMP  003250R     025 M.ABS  = 000000          M.SUBR= 000001
FLONEG  003272R     025 F.LBL  = 000020          INNTM2  005016R     025 M.BOL  = 000200          M.SYM  = 000203
FLSWTB  000222R     025 F.LINE= 000012          INPTR   000064R     025 M.BXAV= 000007          M.SYN  = 000201
FLTLEN= 000010           F.LOC  = 000004          INSNAM= 104462          M.BXCR= 000007          M.TRAN= 000012
FNHEAD  000000RG    020 F.LSTC= 000004          IOERR  = 104477          M.BXCT= 000007          M.$ABS= 000000
FNINSI  005044R     025 F.LUN  = 000042          I.CNT  = 177776          M.BXEX= 000007          M.$CEI= 000000
FORIGN  000032R     025 F.MBCT= 000054          I.DIM  = 177772          M.BXFX= 000007          M.$FLR= 000000
FO.APD= 000106           F.MBC1= 000055          I.LEN  = 000012          M.BXIO= 000007          M.$IDN= 000000
FO.MFY= 000002           F.MBFG= 000056          I.OERR= 000062          M.BXLC= 000003          M.$NEG= 000000
FO.RD  = 000001          F.MON  = 000002          I.OLD  = 177770          M.BXNC= 000007          M.$SIG= 000000
FO.UPD= 000006           F.NIL  = 000001          I.PTR  = 177774          M.BXNL= 000007          M.1RDT= 000011
FO.WRT= 000016           F.NRBD= 000024          I.STR  = 177766          M.BXPP= 000007          M.1REV= 000003
FPMODE  000042R     025 F.NREC= 000030          LABERR= 104406          M.BXPW= 000007          M.1SCN= 000011
FPP    = 000000          F.OPN  = 000010          LABLPT  005042R     025 M.BXRL= 000007          NAMBUF  003240R     025
FPPLEN= 000004           F.OVBS= 000030          LARGBL  003162R     025 M.BXTS= 000007          NB.DEV= 000200
FREBLK  007400R     025 F.PREV= 000000          LARGCV  003176R     025 M.BXWA= 000007          NB.DIR= 000100
FUZZ    000142R     025 F.PRNT= 100000          LARGFL  003160R     025 M.CEI  = 177777          NB.NAM= 000004
FXPTR1  003076R     025 F.PROT= 010000          LARGLE  003174R     025 M.CHR  = 000202          NB.SD1= 000400
F$EVAL= 000100           F.RACC= 000016          LARGLN  003166R     025 M.CMT  = 000206          NB.SD2= 001000
F$EVI  = 000200          F.RATT= 000001          LARGPT  003164R     025 M.DOUT= 000006          NB.SNM= 000040
F$EXEC= 000400           F.RCNM= 000034          LARGSF  003172R     025 M.DQAD= 000006          NB.STP= 000020
F$FUNC= 001000           F.RCTL= 000017          LARGTY  003170R     025 M.ENCD= 000004          NB.SVR= 000010
F.ACTL= 000076           F.READ= 004000          LARGVA  003200R     025 M.EXE  = 000000          NB.TYP= 000002
F.ALOC= 000040           F.REDF= 000040          LBINC   005100R     025 M.EXP  = 000001          NB.VER= 000001
F.BBFS= 000062           F.RES  = 000040          LBUFF   003316R     025 M.FACT= 000001          ND.DEV= 000200
F.BDB  = 000070          F.RET  = 177772          LBUFLN= 000621          M.FIL  = 000177          ND.EXT= 000002
F.BGBC= 000057           F.RSIZ= 000002          LCOUNT  005106R     025 M.FLR  = 177777          ND.NAM= 000004
F.BITS= 177774           F.RTYP= 000000          LENERR= 104412          M.FMT  = 000004          ND.PPN= 000100
F.BKDN= 000026           F.R2   = 000014          LENTAB  000054R     025 M.GDUP= 000003          ND.$EX= 000000
F.BKDS= 000020           F.R3   = 000016          LEX    = ****** GX          M.GDWN= 000003          ND.$E$= 000020
F.BKEF= 000050           F.R4   = 000020          LEXPTR  003100R     025 M.GETI= 000006          ND.$NM= 000000
F.BKP1= 000051           F.SEQN= 000100          LFTOP   005066R     025 M.IBM  = 000003          ND.$N$= 000040
F.BKST= 000024           F.SPDV= 000072          LINC    005102R     025 M.IDN  = 000000          ND.$PN= 001400
F.BKVB= 000064           F.SPUN= 000074          LINE    005016R     025 M.INSP= 000006          NEGSGN  005063R     025
F.BR   = 000020          F.STBK= 000036          LININC  005026R     025 M.IOTA= 000000          NINPT1  003302R     025
F.CHR  = 000075          F.STK  = 000022          LISIZE= 000020          M.LIT  = 000205          NINPT2  003300R     025
F.CNTG= 000034           F.STOP= 000004          LITOP   005072R     025 M.LN   = 000001          NINSTK  003304R     025
F.CODE= 000010           F.STP  = 000001          LNLNG  = 104541          M.MMD  = 000010          NINTMP  003310R     025
F.DFNB= 000046           F.SUSP= 000002          LOCLCT  005036R     025 M.NEG  = 000000          NINTYP  003306R     025
F.DSPT= 000044           F.TRC  = 000002          LOWCOR  000002R     025 M.NOT  = 000000          NOTAVL= 104472
F.DVNM= 000134           F.UNIT= 000136          LSTERP  004766R     025 M.NUB  = 000006          NOTOPN= 104473
F.EFBK= 000010           F.URBD= 000020          LWIDTH  000070R     025 M.NUM  = 000204          NPRT1   004762R     025
F.EFN  = 000050          F.VAL  = 177776          L.BITS= 000003          M.OUTR= 000012          NULL    007364R     025
F.END  = 002000          F.VBN  = 000064          L.CNT  = 000002          M.PITM= 000000          NUMCHN= 000020
F.EOBB= 000032           F.VBSZ= 000060          L.PRNT= 000001          M.PRDS= 000006          NUMDFL  003230R     025
F.ERR  = 000052          GETFRA= ****** GX          L.PTR  = 000000          M.QQAD= 000000          NUMEFL  003233R     025
F.FACC= 000043           GETNXT= ****** GX          L.PTR1= 000004          M.QUAD= 000000          NUMESN  003236R     025
F.FFBY= 000014           GNOME   003102R     025 L.SUBS= 000002          M.RAND= 000012          NUMFLG  003230R     025
F.FLAG= 000002           HASHC   005136R     025 MACHIN= 000106          M.RAVL= 000000          NUMLEN  000062R     025
F.FLB  = 000100          HASHTB  005140R     025 MAXCHN= 000014          M.RCIP= 000000          NUMNEG  003232R     025
F.FNAM= 000110           HSHMSK= 177601          MONBLK  003120R     025 M.RDCT= 000011          NUMQFL  003234R     025
```

SYMBOL TABLE

```
NUMRIG  003231R     025 RANERR= 104411          R.NEXT= 000006       SY.BS = 000010       T.LIT = 000020
NUMWID  003226R     025 RARGBL  003204R   025 R.OFF = 000000       SY.CLR= 000040       T.MON = 000010
NUMZFL  003235R     025 RARGCV  003220R   025 R.PREV= 000004       SY.ERP= 000020       T.NIL = 000004
NYIERR= 104432          RARGFL  003202R   025 R.PTR = 000000       SY.NOU= 000100       T.NUM = 002000
N.BLSZ= 000036          RARGLE  003216R   025 R.SEQ = 000003       SY.OPN= 000002       T.OPFN= 000001
N.DESC= 000020          RARGLN  003210R   025 R.VAR = 000002       SY.PMT= 001000       T.OTHR= 040000
N.DEV = 000032          RARGPT  003206R   025 R6    =%000006       SY.SRT= 000400       T.SOP = 000002
N.DEVN= 000034          RARGSF  003214R   025 R7    =%000007       SY.SVC= 000004       T.SPOP= 000040
N.DID = 000024          RARGTY  003212R   025 SI    =%000005       SY.SVL= 000200       T.TMP = 000200
N.DVNM= 000032          RARGVA  003222R   025 SIDAMN= 104433       S.AB  = 177776       T.VAL = 100000
N.EXT = 000014          RBASE   005074R   025 SIFRAM  003314R 025 S.ASB = 100000       T.VAR = 001000
N.FID = 000000          RCOUNR  005112R   025 SISTAK  006350R 025 S.BFHD= 000020       UPPER = 177400
N.FNAM= 000006          RCOUNT  005110R   025 SITOP   006346R 025 S.BITS= 000002       UTNAME  000344R     023
N.FTYP= 000014          REDCNT  005070R   025 SITP    007144R 025 S.CHR1= 000003       VALERR= 104413
N.FVER= 000016          REDIGN  005066R   025 SIWARN= 000200       S.FATT= 000016       VARBLE  007400R     025
N.NAME= 000006          REDOFF  005072R   025 SIZTAB  000044R 025 S.FDB = 000140       VARBLS  000000R     025
N.NEXT= 000022          REENTR  003060R   025 SI.BIT= 177774       S.FLB = 040000       VERNUM  000000R     025
N.PPN = 000024          RESBIT  003142R   025 SI.LEN= 177772       S.FNAM= 000006       VRSION= 000002
N.STAT= 000020          RESCVT  003152R   025 SI.RET= 177772       S.FNB = 000036       WIDTH   000072R     025
N.UNIT= 000034          RESERR= 104440          SI.VAL= 177776       S.FNBW= 000017       WROUND  000020R     025
OBUFLN= 001000          RESLEN  003150R   025 SKPSIZ  005134R 025 S.FNTY= 000004       WSDATE  000136R     025
OCHAN   005052R     025 RESPTR  003140R   025 SPHOLD  000004R 024 S.FTYP= 000002       WSFULL= 104403
OMODE   000402R     023 RESSFT  003146R   025 SPINIT  000002R 024 S.LEN = 000002       WSNAME  000074R     025
OPTMP0  005066R     025 RESTYP  003144R   025 STKINC= 000774       S.LINK= 177774       WSTIME  000132R     025
OPTMP1  005070R     025 RESULT  003136R   025 STOP  = 104420       S.NFEN= 000020       $CLRF = ****** GX
OPTMP2  005072R     025 RESVAL  003154R   025 SUBCNT  003104R 025 S.TYPE= 140000       $FIS  = ****** G
OPTMP3  005074R     025 RETCOR= ****** GX       SUBDCT  003106R 025 S.VAL = 000000       $FPP  = ****** G
OPTMP4  005076R     025 RINC    005104R   025 SUBPTR  003110R 025 TKDIMS  005072R 025 $FPPLN= ****** G
OPTMP5  005100R     025 RLTBAS  005066R   025 SYMB1   005344R 025 TKFRAM  005066R 025 $MACHN= ****** G
OPTMP6  005102R     025 ROTINC  005070R   025 SYMF    005340R 025 TKLEFT  005070R 025 $RSTS = ****** G
OPTMP7  005104R     025 ROUND   000010R   025 SYMPT1  005346R 025 TMODE   000403R 023 $RSX  = ****** G
OPTMP8  005106R     025 ROVER   003312R   025 SYMTAB  005352R 025 TMPTAG= 000166       $RT11 = ****** G
OPTMP9  005110R     025 RSTS  = 000000          SYMTBF= 104404       TTYBUF  004140R 025 .BOOLE= 000002
OPTM10  005112R     025 RSX   = 000115          SYMTOP  005350R 025 TTYLEN= 000622       .CHARA= 000000
OPTM14  005026R     025 RTOP    005070R   025 SYMTP   006272R 025 TURKEY= 104436       .DEF. = 000002
ORIGIN  000030R     025 RT11  = 000000          SYMWRN= 000054       TYPE  = 000166       .FLOAT= 000006
OUTPTR  000066R     025 R.BITS= 000003          SYNERR= 104407       T.BOL = 020000       .FSRCB= ****** G
OVROPR= 000000          R.FIX = 000001          SYSERR= 104402       T.CON = 004000       .INTEG= 000004
PATCH   007344R     025 R.LCT = 000002          SYSTAT  000000R 024 T.DIM = 000100       .MOLUN  000404R     023
PRECSN  000052R     025 R.LN  = 000002          SY.BAD= 000001       T.FUN = 010000       ...GBL= 000000
RANDAC  003064R     025 R.LNUM= 000010          SY.BRK= 100000       T.LIST= 000400       ...TPC= 000020
RANDY   003062R     025

. ABS.  000000  000
        000000  001
TABLES  000000  002
MONTB   000000  003
DYDTB   000000  004
BOPTB   000000  005
MONBTB  000000  006
DYDBTB  000000  007
COMTB   000000  010
CHRTBL  000000  011
VTBL    000000  012
ASCTBL  000000  013
OVRTBL  000000  014
TRNTBL  000000  015
```

```
FNHEAD  MACRO M1113  08-FEB-80 12:30  PAGE 47-4
SYMBOL TABLE

MNTBL   000000      016
MNTRTL  000000      017
CODE    000660      020
PATCH   000000      021
$$FSR1  000000      022
DEF$0   000406      023
DEF$1   000006      024
DEF$2   007400      025
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  13028 WORDS  ( 51 PAGES)
DYNAMIC MEMORY:  14478 WORDS  ( 55 PAGES)
ELAPSED TIME:  00:02:38
FNHEAD,DM1:[305,301]FNHEAD/-SP=APLRX1,APLDEF,FNHEAD,END
```

B 1
C 1
D 1
E 1 ERASE
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1
B 2
C 2
D 2 ERASEC
E 2
F 2
G 2
H 2
I 2
J 2
K 2
L 2
M 2
N 2
B 3
C 3 ERRMSG
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4 ERROR
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5
I 5
J 5
K 5 EXEC
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7 EXISTS
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7 EXITIT
N 7
B 8
C 8
D 8
E 8
F 8
G 8
H 8
I 8
J 8
K 8 EXPR
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9
J 9
K 9
L 9
M 9 FIXLIN
N 9
B 10
C 10
D 10
E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11 FN
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13 FNEND
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14 FNEXEC
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14
M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15 FNHEAD
N 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16